

Algoritmo Eficiente de Agrupación de Subespacios Sobre Imágenes Espectrales Utilizando  
Muestreo Restringido por Similitud

Jhon Edinson López Durán

Trabajo de Grado para optar al título de Ingeniero de Sistemas

Director

Carlos Alberto Hinojosa Montero

Magíster en ingeniería de Sistemas

Codirector

Henry Arguello Fuentes

Doctorado en Ingeniería Eléctrica y Computación

Universidad Industrial de Santander

Facultad de Ingenierías Fisicomecánicas

Escuela de Ingenierías de Sistemas e Informática

Bucaramanga

2020

### **Dedicatoria**

A mi madre,

A mi padre,

A mi hermano,

A Dante.

### **Agradecimientos**

Al Magíster Carlos Alberto Hinojosa Montero, director del proyecto, por ser mi mentor, brindarme su acompañamiento a lo largo del desarrollo de este proyecto.

A Mayra y Layoneth, mis mejores amigos, por su apoyo incondicional durante todos estos años.

## Tabla de Contenido

<b>Introducción</b>	<b>15</b>
<b>1. Antecedentes teóricos</b>	<b>20</b>
1.1. Imágenes espectrales	20
1.2. Teoría de la agrupación de subespacios	23
1.2.1. Algoritmo escalable de agrupación de subespacios usando muestreo aleatorio	29
1.3. Segmentación de imágenes	33
1.3.1. Detección de bordes y umbralización	33
1.3.2. Super píxeles	37
<b>2. Método Propuesto</b>	<b>40</b>
2.1. Algoritmo de agrupación de subespacios con restricción de escasez (SSC)	41
2.2. Detección de bordes y superpíxeles	42
2.3. Muestreo restringido por similitud	43
2.4. Análisis de Complejidad Computacional	48
<b>3. Simulaciones y resultados</b>	<b>50</b>
3.1. Análisis del parámetro $\phi$	53

AGRUPACIÓN DE SUBESPACIOS UTILIZANDO MUESTREO RESTRINGIDO POR SIMILITUD	7
3.2. Mapas Visuales y Resultados Cuantitativos	55
3.3. Análisis del rendimiento en función de $\rho$	59
<b>4. Conclusiones</b>	<b>64</b>
<b>Referencias Bibliográficas</b>	<b>65</b>
<b>Apéndices</b>	<b>72</b>

### Lista de Figuras

Figura 1.	Representación de la información espectral en HSI (Velasco et al., 2015).	16
Figura 2.	Concepto de Imagen Espectral (Hinojosa Montero, 2018).	22
Figura 3.	Clasificación de imágenes espectrales utilizando el enfoque de agrupamiento de subespacios (Hinojosa Montero, 2018).	28
Figura 4.	Histograma bimodal. $T$ es el valor de intensidad umbral y $t_0$ y $t_1$ son los dos posibles niveles de gris de los píxeles de la imagen umbralizada (Osorio et al., 2011).	37
Figura 5.	Método de agrupación de subespacios propuesto.	41
Figura 6.	(a) Representación en escala de grises de una HSI. (b) Resultado de la detección de bordes mediante la suma de todas las bandas espectrales de la HSI y luego aplicar superpíxeles. (c) Suma de todas las bandas de la HSI y calcular superpíxeles.	43
Figura 7.	Selección de puntos dentro y fuera de la muestra.	44
Figura 8.	Indian Pines. (a) Imagen en falso color. (b) Imagen de etiquetas de las clases presentes. (c) Curvas espectrales de todas las clases.	52
Figura 9.	Salinas. (a) Imagen en falso color. (b) Imagen de etiquetas de las clases presentes. (c) Curvas espectrales de todas las clases.	53
Figura 10.	Universidad de Pavia. (a) Imagen en falso color. (b) Imagen de etiquetas de las clases presentes. (c) Curvas espectrales de todas las clases.	54

Figura 11.	Influencia del parámetro $\phi$ en la precisión general del agrupamiento en la imagen Salinas.	55
Figura 12.	Resultado de agrupamiento en Indian Pines . (a) Imagen de etiquetas de las clases presentes . (b) SRSSSC con $N_{seg} = 400$ . (c) SR-SCC. (d) SSSC.	59
Figura 13.	Resultado de agrupamiento de la universidad de Pavia. (a) Imagen de etiquetas de las clases presentes . (b) SRSSSC con $N_{seg} = 650$ . (c) SR-SCC. (d) SSSC.	60
Figura 14.	Resultado de agrupamiento Salinas.(a) Imagen de etiquetas de las clases presentes . (b) SRSSSC con $N_{seg} = 600$ . (c) SR-SCC. (d) SSSC.	62
Figura 15.	Resultado de agrupamiento Indian Pines calculando el valor de $k$ sobre cada segmento. (a) Precisión general (OA). (b) Tiempo de ejecución.	62
Figura 16.	Resultado de agrupamiento Indian Pines.	63
Figura 17.	Resultado de agrupamiento universidad de Pavia.	63
Figura 18.	Resultado de agrupamiento Salinas.	63

### Lista de Tablas

Tabla 1.	Complejidad temporal y espacial del algoritmo propuesto, donde $t_1$ y $t_2$ son las iteraciones necesarias para resolver el problema de optimización mostrado en 21.	48
Tabla 2.	Complejidad computacional de SSC, SSSC, y el algoritmo propuesto en este trabajo. $t_1$ corresponde al número de iteraciones necesarias para la convergencia de la norma $\ell_1$ en el problema de optimización 21. $t_2$ , son las iteraciones para el algoritmo k-means.	49
Tabla 3.	Resultados cuantitativos de agrupación en la imagen Indian Pines	57
Tabla 4.	Resultados cuantitativos de agrupación en la universidad de Pavia.	58
Tabla 5.	Resultados cuantitativos de agrupación en la imagen Salinas.	61

### Lista de Apéndices

	<b>pág.</b>
Apéndice A. ADMM para resolver el problema de optimización del agrupamiento de subespacio propuesto	72
Apéndice B. Métricas de Evaluación	77

**Abreviaturas empleadas**

Notación	Definición
$H$	Imagen hiperespectral (HSI)
$p$	Número de puntos dentro de la muestra
$k$	Número de subespacios presentes
$M$	Número de filas de la HSI
$N$	Número de columnas de la HSI
$L$	Número de Bandas espectrales de la HSI
$n$	Número de puntos de datos ( $M \cdot N$ )
$N_{seg}$	Cantidad de superpíxeles
$\mathbf{S}_p = [\mathbf{s}_{p1}, \mathbf{s}_{p2}, \dots, \mathbf{s}_{pN_s}]$	Imagen representada con superpíxeles
$\mathbf{B}$	Imagen binaria resultante de la detección de bordes
$\mathbf{D} = [\mathbf{d}_1, \mathbf{d}_2, \dots, \mathbf{d}_n]$	Conjunto de datos
$\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_p]$	Puntos dentro de la muestra
$\mathbf{Y} = [\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_{n-p}]$	Puntos fuera de la muestra
$\mathbf{C} = [\mathbf{c}_1, \mathbf{c}_2, \dots]$	Representación escasa del conjunto de datos
$\mathbf{A} \in \mathbb{R}^{n \times n}$	Matriz de afinidad obtenida a partir de $\mathbf{C}$
$\mathbf{L} \in \mathbb{R}^{n \times n}$	Matriz Laplaciana
$\mathbf{V} \in \mathbb{R}^{n \times k}$	Matriz de autovalores
$\mathbf{Ac}$	Matriz de etiquetas inicializada en ceros.
$\mathbf{K} = [k_1, k_2, \dots, k_{N_{seg}}]$	Vector que contiene la cantidad de grupos presentes en cada segmento.

## Resumen

**Título:** Algoritmo Eficiente de Agrupación de Subespacios Sobre Imágenes Espectrales Utilizando Muestreo Restringido por Similitud \*

**Autor:** Jhon Edinson López Durán \*\*

**Palabras Clave:** Imágenes espectrales, Muestreo, Clasificación no Supervisada, Agrupamiento de Subespacios.

**Descripción:** La precisión de los métodos que abordan la agrupación de subespacios mediante la agrupación espectral, dependen de la construcción de una matriz de afinidad para crear el grafo de similitud de los puntos. Algunos trabajos como la agrupación de subespacios mediante la restricción de escasez (SSC, por sus siglas en inglés) han logrado resultados eficientes a la hora de asignar un punto a un subespacio. Sin embargo estos métodos presentan limitaciones cuando la cantidad de puntos a agrupar aumenta, ya que la complejidad temporal crece a una razón cubica del tamaño de los datos, lo que hace que estos métodos sean ineficientes a la hora de abordar problemas de gran escala, como el agrupamiento de subespacios en imágenes espectrales. Este trabajo de investigación busca disminuir la complejidad computacional temporal y espacial a la hora de aplicar agrupamiento espectral en imágenes espectrales, dividiendo la data en segmentos y aplicando sobre ellos un muestreo teniendo en cuenta la información espacial, para obtener puntos dentro de la muestra y puntos fuera de la muestra. A cada subconjunto se le aplicará un tratamiento diferente para así lograr la asignación de grupos de todos los puntos. Además se presentan resultados experimentales utilizando varias imágenes espectrales, los cuales muestran la efectividad de este enfoque.

---

\* Trabajo de grado

\*\* Facultad de Ingenierías Físico-Mecánicas. Escuela de Ingeniería de Sistemas e Informática. Director: Carlos Alberto Hinojosa Montero, Magíster en ingeniería de Sistemas.

## Abstract

**Title:** Sampling Restricted by Similarity for Sparse Subspace Clustering \*

**Author:** Jhon Edinson López Durán \*\*

**Keywords:** Spectral Imaging, Sampling, Unsupervised Learning, Clustering, Sparse Subspace Clustering.

**Description:** The precision of the methods that address the subspace clustering by means of the spectral cluster, depend on the construction of an affinity matrix to create the similarity graph of the points. Some works such as sparse subspace clustering (SSC) have achieved efficient results when assigning a point to a subspace. However, these methods have limitations when the number of points to be grouped increases, since the temporal complexity grows at a cubic ratio of the size of the data, which makes these methods inefficient in addressing large-scale problems, such as the grouping of subspaces in spectral images. This research work seeks to reduce the temporal and spatial computational complexity when applying spectral clustering in spectral images, dividing the data into segments and applying a sampling taking into account the spatial information, to obtain data in-sample and data out-of-sample. A different treatment will be applied to each subset in order to achieve the group assignment of all points. In addition, experimental results are presented using several spectral images, which show the effectiveness of this approach.

---

\* Research Work

\*\* School of Physical-Mechanical Engineering. Department of Systems Engineering and Informatics. Advisor, Carlos Alberto Hinojosa Montero. Master in Systems Engineering.

## Introducción

Los dispositivos de adquisición de imágenes espectrales capturan la información espectral de una escena al obtener una gran cantidad de información espacial en diferentes frecuencias de radiación electromagnética. Las imágenes espectrales se consideran conjuntos de datos tridimensionales o cubos de datos con dos dimensiones en el dominio del espacio  $(x,y)$  y uno en el dominio de la longitud de onda  $(\lambda)$ . En general, las técnicas de detección tradicionales construyen un cubo de datos espacio-espectral explorando la escena, ya sea espectral o espacialmente en proporción a la resolución espacial o espectral deseada, aumentando así los tiempos de adquisición. Por otro lado, los dispositivos de adquisición de una sola captura obtienen la información espacial y espectral de una escena mediante el mapeo de todos los vóxeles del cubo de datos espectrales en diferentes regiones del vector del plano focal (**FPA**, por sus siglas en inglés) (Gao et al., 2010; Kerekes and Schott, 2007)

En general, las técnicas tradicionales requieren detectar cada vóxel de la escena 3D, por lo tanto, se necesitan grandes capacidades de almacenamiento y recursos de cómputo para almacenar y procesar tales imágenes de alta dimensión.

El conocimiento del contenido espectral en varias ubicaciones espaciales desde una escena puede ser una herramienta valiosa para la detección, identificación y clasificación de materiales y objetos con composiciones complejas (He et al., 2015). En particular la clasificación de imágenes espectrales es una importante tarea para muchas aplicaciones prácticas, cómo la precisión en la

agricultura, seguimiento y gestión del medio ambiente, situaciones de seguridad y defensa. Como se muestra en la figura 1, cada posición espacial en una imagen espectral es representada por un vector, donde estos valores corresponde a la intensidad de diferentes bandas espectrales. Estos vectores también se conocen como la firma espectral de los píxeles. Por lo general, diferentes materiales reflejan la energía electromagnética de manera diferente en longitudes de onda específicas (Shaw and Burke, 2003), de ahí que la información proporcionada por cada ubicación espacial en una imagen espectral esté representada por un vector cuyos valores corresponden a la intensidad en diferentes bandas espectrales, lo cual permite distinguir diferentes materiales lo que conlleva a obtener una clasificación en imágenes espectrales más precisa.

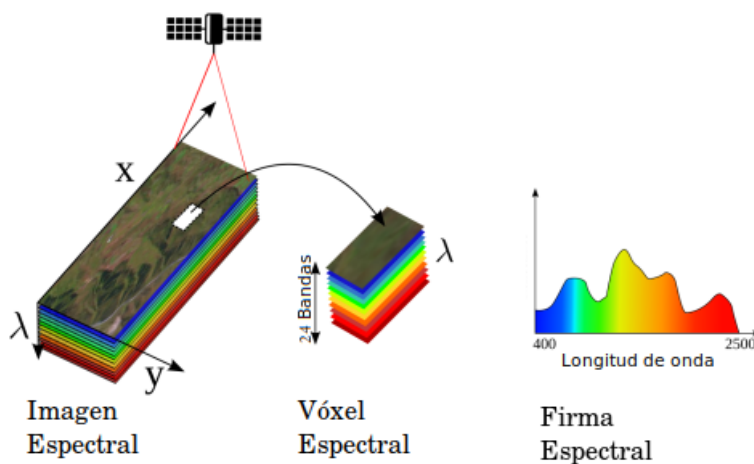


Figura 1. Representación de la información espectral en HSI (Velasco et al., 2015).

En general, existen dos enfoques principales para el problema de clasificación: el enfoque supervisado y no supervisado. Las técnicas supervisadas requieren la disponibilidad de un conjunto de entrenamiento para que el clasificador pueda aprender. Entre las diversas técnicas supervisadas, existen las máquinas de soporte vectorial (SVM, por sus siglas en inglés) (Boser et al., 1992; Vap-

nik, 2013). Los métodos no supervisados, entre estos los métodos de agrupación de subespacios (en inglés, **subspace clustering**) y análisis de grupos, realizan la clasificación explotando la información inherente de los datos, sin requerir ningún conjunto de muestras de entrenamiento.

El análisis de grupos (en inglés, **cluster analysis**) es una técnica de clasificación común que busca agrupar un conjunto de datos de tal manera que los miembros del mismo grupo (llamado clúster) sean más similares entre sí. El análisis de grupos no es en sí un algoritmo específico, sino la tarea a realizar. Se puede hacer el agrupamiento utilizando varios algoritmos que difieren significativamente en su idea de qué constituye un grupo y cómo encontrarlos eficientemente. Las ideas clásicas de grupo incluyen distancias pequeñas entre los miembros del mismo, áreas densas del espacio de datos, intervalos o distribuciones estadísticas particulares. El agrupamiento, por tanto, puede ser formulado como un problema multiobjetivo de optimización. El algoritmo apropiado y los valores de los parámetros (incluyendo valores como la función de distancia para utilizar, un umbral de densidad o el número de grupos esperado) depende del conjunto de datos que se analiza y el uso que se le dará a los resultados. Los algoritmos de agrupamiento de subespacios se pueden dividir en cuatro categorías principales: iterativos (Zhang et al., 2009), algebraicos (Vidal et al., 2005), estadísticos (Ma et al., 2007; Tipping and Bishop, 1999) y aquellos basados en agrupamiento espectral (en inglés, **spectral clustering**). En particular, la agrupación espectral encuentra la pertenencia de los puntos de datos a un grupo utilizando el espectro de una matriz de afinidad no negativa simétrica cuyas entradas miden las similitudes entre los puntos conectados. Por lo tanto, el paso más importante en los métodos basados en agrupación espectral es el problema

de construcción del grafo de similitud (Ng et al., 2002). En los últimos años, un nuevo algoritmo denominado agrupación de subespacios con restricción de escasez (**SSC**, por sus siglas en inglés) propone capturar la relación geométrica global entre todos los puntos de datos expresando cada punto de datos como una combinación lineal de todos los demás puntos y, a continuación, el conjunto de soluciones se restringe a ser disperso o escaso, minimizando la norma  $\ell_1$  de la matriz de coeficientes de representación. Usando la matriz de representación dispersa, se construye un grafo de similitud, a partir del cual se obtiene la segmentación de los datos (Elhamifar and Vidal, 2009, 2013).

Suponiendo que las firmas espectrales con características espectrales similares se encuentran en el mismo subespacio de baja dimensión, la teoría de agrupación de subespacios puede utilizarse para modelar el problema de agrupación de imágenes espectrales. Recientemente en (Zhang et al., 2016) se ha propuesto un nuevo algoritmo basado en SSC para imágenes espectrales. Específicamente, se propone una variación del algoritmo SSC que considera las propiedades espectrales y espaciales de HSI, la alta correlación espectral y la rica información espacial de HSI, para obtener una matriz de coeficientes más precisa. Aunque este algoritmo de vanguardia logra resultados de alta precisión, los pasos adicionales de procesamiento requeridos para extraer la información espacial y espectral de la escena aumentan el costo computacional. En general, el agrupamiento de imágenes espectrales es una tarea muy desafiante debido a la complejidad inherente de los datos y al costo computacional, que crece en proporción a las dimensiones de los conjuntos de datos espectrales. Además, cuando la relación entre las bandas espectrales y el número de muestras de datos es muy diferente, los datos de alta dimensión sufren del fenómeno conocido como la "maldición

de dimensionalidad" (Steinbach et al., 2004).

Con el objetivo de realizar una agrupación de subespacios más eficiente, diferentes trabajos han propuesto enfoques escalables. Por ejemplo, en (Peng et al., 2013), se aborda el problema de agrupamiento de subespacios dividiendo el conjunto de datos aleatoriamente en dos subconjuntos. Al primer subconjunto se le aplica un algoritmo de agrupación de subespacios, como SSC, donde el resultado de agrupamiento será utilizado para luego agrupar el subconjunto restante. Cada punto del segundo subconjunto se proyecta sobre cada uno de los subespacios calculados anteriormente y se calcula el residuo o la diferencia de cada punto sobre cada subespacio y se asigna al subespacio en el cual haya obtenido el menor residuo. Teniendo en cuenta que para dividir los datos en dos subconjuntos se utiliza un enfoque aleatorio y los algoritmos de agrupamiento de subespacios presentan inconvenientes a la hora de agrupar puntos cercanos a dos subespacios distintos; esta propuesta de investigación propone diseñar un enfoque escalable de agrupamiento de subespacios sobre imágenes espectrales aplicando algoritmos de segmentación de imágenes tales como detección de bordes y superpíxeles para así lograr dividir de una manera eficiente los datos y mejorar la precisión de clasificación.

## **1. Antecedentes teóricos**

En este capítulo se introducen los antecedentes teóricos relacionados con el agrupamiento subespacial en imágenes espectrales. Primero, se presentan el concepto de imágenes espectrales y sus características. En la segunda parte, se presentan brevemente la teoría de la agrupación de subespacios y algunos algoritmos. Seguido por la descripción de los algoritmos utilizados para generar el muestreo restringido por similitud y finalmente se mencionan algunos métodos para la segmentación de imágenes.

### **1.1. Imágenes espectrales**

El sistema visual humano construye una representación del entorno circundante al detectar e interpretar la información del rango visible, aproximadamente de 380 a 750 nm, del espectro electromagnético. Esta limitación se extiende a las cámaras fotográficas tradicionales que agrupan toda la información espectral dentro del rango espectral visible en tres amplios rangos espectrales que corresponden aproximadamente a los tres colores primarios, rojo, verde y azul. Aunque dicha información permite percibir la forma, la textura de la superficie y la relación espacial mutua en la profundidad del espacio 3D, es insuficiente en aplicaciones donde la información espectral de interés se extiende a otros rangos del espectro electromagnético.

Las imágenes espectrales combinan la espectroscopia y las metodologías para la formación de imágenes bidimensionales. Mientras que las imágenes proporcionan la intensidad en cada píxel de una imagen 2D, y un espectrómetro típico mide un solo espectro, las imágenes espectrales recopilan imágenes 2D en longitudes de ondas específicas a través del espectro electromagnético.

Está información puede representarse como un conjunto de datos tridimensionales (3D).

Cada ubicación espacial en una imagen espectral está representada por un vector cuyos valores corresponden a la intensidad en diferentes bandas espectrales. Estos vectores también se conocen como la firma espectral de los píxeles, ver figura 2. Todos los materiales tienen características espectrales únicas porque absorben, reflejan y emiten radiación de una manera única. Por ejemplo, en la parte visible del espectro, una hoja aparece verde porque se absorbe en la región azul y roja del espectro y se refleja en la región verde. Estas variaciones en la absorción, reflexión y emisión se deben a la composición del material. Las diferencias en la respuesta espectral debido a la absorción, transmisión y reflexión hacen que los materiales tengan una firma espectral única. Por lo tanto, la información proporcionada por las firmas espectrales puede ser una herramienta valiosa para la detección, identificación y clasificación de materiales y objetos con composiciones complejas(He et al., 2015). En particular, la clasificación de imágenes espectrales consiste en etiquetar firmas espectrales individuales en una de las clases según sus características espectrales. La clasificación de imágenes espectrales ha encontrado muchas aplicaciones en varios campos, como el militar (Manolakis and Shaw, 2002; Eismann et al., 2009), agricultura de precisión (Patel et al., 2001), y mineralogía (Hörig et al., 2001).

Constantemente se realizan investigaciones para desarrollar métodos avanzados de clasificación y segmentación de imágenes, y técnicas para mejorar la precisión, sin embargo clasificar los datos presentes en las imágenes espectrales es un tema complejo debido a que se tiene que considerar que la información presente en la imagen espectral proviene de objetos diversos. De acuerdo con el proceso de construcción, los métodos de clasificación de imágenes se dividen en

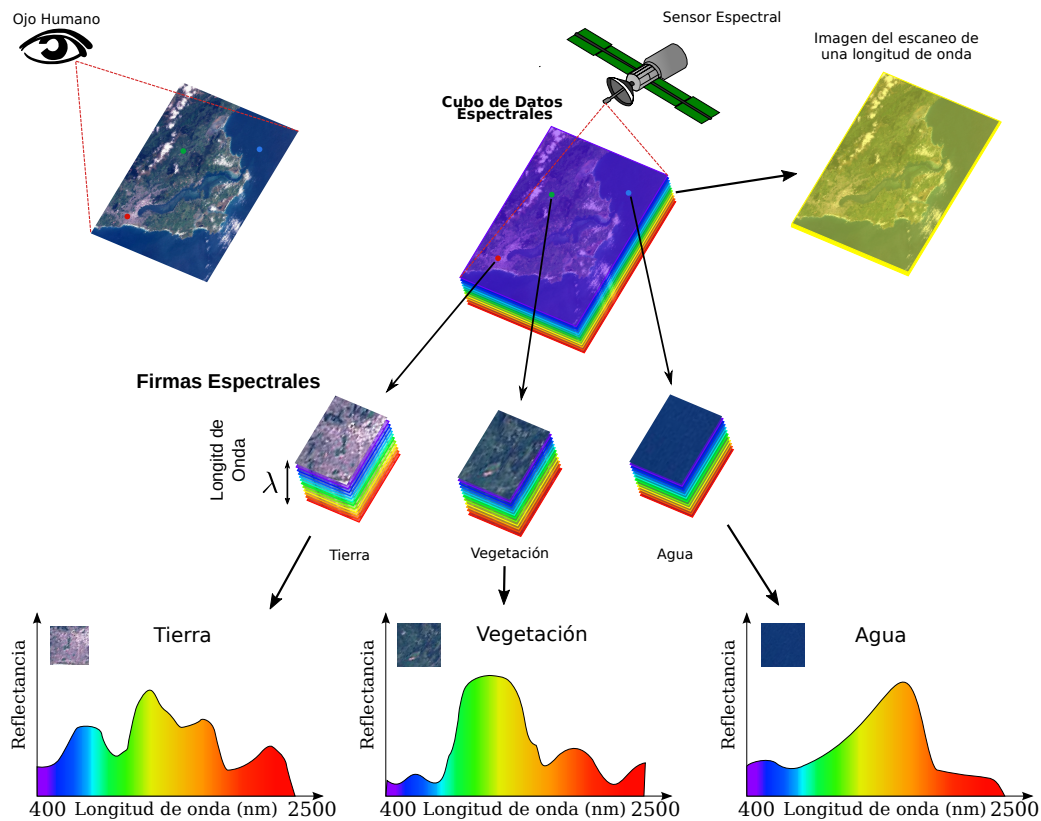


Figura 2. Concepto de Imagen Espectral (Hinojosa Montero, 2018).

dos grupos(?): **Supervisado**: es un método de reconocimiento de patrones basado en la existencia de un conjunto de prototipos predefinidos; **No supervisado**: no requiere conocimiento previo de la zona de estudio y desarrolla la clasificación en forma automática, aprovechando las características espectrales de la imagen para definir las agrupaciones con valores similares. Los algoritmos de clasificación no supervisada se subdividen en dos grupos: con número de clases conocido y desconocido. El primer grupo incluye el método de agrupación de subespacios tales como el K-means. En los métodos con número de clases desconocidas se encuentran el adaptativo y el de Batchelor y Wilkins, conocido como método de máxima distancia (Macedo-Cruz et al., 2010). Luego, el objetivo es estimar simultáneamente estos subespacios y agrupar los puntos en sus subespacios

correspondientes.

## 1.2. Teoría de la agrupación de subespacios

En los últimos años, los avances tecnológicos sin precedentes han llevado a un incremento en la disponibilidad y dimensionalidad de los datos en todas las áreas de la ciencia y la ingeniería. Estos incluyen aprendizaje automático, procesamiento de señales e imágenes, visión por computadora, reconocimiento de patrones, bioinformática, etc. Por ejemplo, una imagen de escala de grises convencional consta de miles de millones de píxeles, mientras que una imagen espectral consta de cientos de imágenes en escala de grises que proporcionan información de diferentes longitudes de onda. Esta alta dimensionalidad de los datos conduce a un incremento en los recursos de memoria, con el fin de almacenar dicha información, y el costo computacional para el procesamiento y análisis de datos. Sin embargo, los datos de alta dimensión a menudo se encuentran en estructuras de baja dimensión en lugar de estar distribuidos uniformemente a través del espacio. Por lo tanto, se han desarrollado diferentes técnicas para encontrar una representación de baja dimensión de un conjunto de datos de alta dimensión (Vidal, 2011).

Una técnica tradicional es el análisis de componentes principales (**PCA**, por sus siglas en inglés). PCA asume que un conjunto de puntos  $\{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N\} = \{\mathbf{x}_j\}_{j=1}^N$  en el espacio dimensional alto  $\mathbb{R}^D$  se extrae de un único subespacio afín de baja dimensión  $S$  de dimensión  $d \ll D$ . Esta técnica está bien establecida en la literatura y se ha convertido en una de las herramientas más útiles para el modelado, compresión y visualización de datos (Vidal, 2011).

En la práctica, sin embargo, los puntos de datos podrían ser extraídos de la unión de  $n \geq 1$  subespacios lineales o afines  $\{S_i\}_{i=1}^n$  de dimensiones desconocidas  $d_i = \dim(S_i)$ ,  $0 < d_i < D$ . Los

subespacios se pueden describir como

$$S_i = \{\mathbf{x} \in \mathbb{R}^D : \mathbf{x} = \boldsymbol{\mu}_i + \mathbf{U}_i \mathbf{y}\}, \quad i = 1, \dots, n, \quad (1)$$

dónde  $\boldsymbol{\mu}_i \in \mathbb{R}^D$  es un punto arbitrario en el subespacio  $S_i$ , que puede ser elegido como  $\boldsymbol{\mu}_i = \mathbf{0}$  para subespacios lineales;  $\mathbf{U}_i \in \mathbb{R}^{D \times d_i}$  es una base para el subespacio  $S_i$ ; y  $\mathbf{y} \in \mathbb{R}^{d_i}$  es una representación dimensional baja para el punto  $\mathbf{x}$ . Entonces, el objetivo del agrupamiento subsespacial es encontrar el número de subespacios  $n$ , sus dimensiones  $\{d_i\}_{i=1}^n$ , las bases subsespaciales  $\{\mathbf{U}_i\}_{i=1}^n$ , los puntos  $\{\boldsymbol{\mu}_i\}_{i=1}^n$ , y agrupar los puntos de datos en sus subespacios correspondientes. Dado que los datos en un subespacio a menudo se distribuyen de forma arbitraria y no alrededor de un centroide, los métodos estándar de agrupación basados en centroides (Duda et al., 2012) que aprovechan la proximidad espacial de los datos en cada grupo no son, en general, aplicables a los subespacios.

En las últimas dos décadas se han propuesto diferentes algoritmos para la agrupación de subespacios que tienen en cuenta la estructura multisubespacios de los datos. Estos algoritmos se pueden dividir en cuatro categorías principales: métodos basados en agrupación iterativa, algebraica, estadística y espectral (Elhamifar and Vidal, 2013). En particular, los métodos basados en agrupación espectral construyen un grafo ponderado  $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathbf{W})$ , dónde  $\mathcal{V} = \{1, \dots, N\}$  es el conjunto de nodos,  $\mathcal{E} \subset \mathcal{V} \times \mathcal{V}$  es el conjunto de ejes, y  $\mathbf{W} \in \mathbb{R}^{N \times N}$  es una matriz de afinidad no negativa simétrica cuya  $(j, k)$  primera entrada,  $W_{jk}$ , mide la afinidad entre puntos  $\mathbf{x}_j$  y  $\mathbf{x}_k$ . Idealmente,  $W_{jk} = 1$  si los puntos  $j$  y  $k$  son del mismo grupo  $W_{jk} = 0$  si los puntos  $j$  y  $k$  son de diferentes

grupos. En la práctica, una afinidad típica está dada por

$$W_{jk} = \exp\left(-\frac{1}{2\sigma}\text{dist}(\mathbf{x}_j, \mathbf{x}_k)^2\right), \quad (2)$$

dónde  $\text{dist}(\mathbf{x}_j, \mathbf{x}_k)$  es una medida de la distancia entre puntos  $j$  y  $k$  y  $\sigma > 0$  es un parámetro. Denotando que  $\mathcal{D} = \text{diag}(\mathbf{W}\mathbf{1})$ , dónde  $\mathbf{W}\mathbf{1} \in \mathbb{R}^N$  es un vector de unos, como una matriz diagonal cuya  $j$ ésima entrada de la diagonal da el grado  $d_{jj} = \sum_k W_{jk}$  del nodo  $j$ , y permite  $\mathcal{L} = \mathcal{D} - \mathbf{W} \in \mathbb{R}^{N \times N}$  es el grafo de la matriz Laplaciana. La agrupación espectral obtiene una agrupación de los datos aplicando el algoritmo K-means a las columnas de la matriz  $\mathbf{Y} = [\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_n]^T \in \mathbb{R}^{n \times N}$ , dónde  $\{\mathbf{u}_i\}_{i=1}^n$  son los autovectores  $\mathcal{L}$  asociado con sus  $n$  autovalores más pequeños. Sin embargo, la afinidad basada en la distancia descrita en Eq. (2) no es apropiado para la agrupación de subespacios ya que dos puntos podrían estar muy cerca uno del otro pero estar en diferentes subespacios. A la inversa, dos puntos podrían estar lejos el uno del otro pero estar en el mismo subespacio. En general, las relaciones geométricas entre múltiples puntos deben considerarse para construir una medida de afinidad efectiva para la agrupación en subespacios (Vidal, 2011; Elhamifar and Vidal, 2013).

De hecho, la construcción de una buena matriz de afinidad es uno de los principales desafíos en la aplicación del clustering espectral al problema del cluster subespacial. Basándose en el método representativo para diseñar una matriz de afinidad, los algoritmos basados en agrupamiento espectral existentes se pueden clasificar en métodos locales y globales. Métodos locales como la afinidad de subespacio local (**LSA**, por sus siglas en inglés) (Yan and Pollefeys, 2006), agrupa-

miento localmente de colectores lineales (**LLMC**, por sus siglas en inglés) (Goh and Vidal, 2007) y planos de mejor ajuste local espectral (**SLBF**, por sus siglas en inglés) (Zhang et al., 2012; Zelnik-Manor and Irani, 2003) calcula una afinidad entre dos puntos que depende solo de los puntos de datos en un vecindario local de cada uno de los dos puntos. Estos métodos tienen dificultades para tratar los puntos cerca de dos subespacios, porque la vecindad de un punto puede contener puntos pertenecientes a subespacios diferentes. Además, son sensibles a la elección correcta del tamaño del vecindario para calcular la información local en cada punto. Por otro lado, los métodos globales como la agrupación de curvatura espectral (**SCC**, por sus siglas en inglés) (Chen and Lerman, 2009) y afinidad del subespacio algebraico (Ng et al., 2002) calcula una afinidad entre dos puntos que depende de todos los puntos de datos.

Recientemente, se ha propuesto un nuevo algoritmo basado en agrupamiento espectral llamado agrupación de subespacios con restricción de escasez (**SSC**, por sus siglas en inglés) (Elhamifar and Vidal, 2009, 2013). SSC también se basa en la idea de escribir un punto de datos como una combinación lineal o afín de puntos de datos vecinos. Sin embargo, mientras LSA, SLBF y LLMC usan la distancia angular o euclidiana entre dos puntos para elegir los  $k$  vecinos más próximos, SSC utiliza el principio de escasez para elegir cualquiera de los puntos de datos restantes ( $N - 1 \gg K$ ) como un posible vecino. Específicamente, se escribe la matriz que contiene todos los puntos de datos sin ruido como:

$$\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N] = [\mathbf{X}_1, \dots, \mathbf{X}_n] \Gamma, \quad (3)$$

dónde  $\mathbf{X}_i \in \mathbb{R}^{D \times N_i}$  es una matriz de rango- $d_i$  de los puntos  $N_i > d_i$  se encuentra en  $S_i$ , y  $\Gamma \in \mathbb{R}^{N \times N}$  es una matriz de permutación desconocida. El algoritmo SSC aprovecha la propiedad de auto expresividad de los datos, i.e., cada punto de datos en una unión de subespacios puede reconstruirse de manera eficiente mediante una combinación de otros puntos en el conjunto de datos. Luego con el  $\mathbf{X}$  La propia matriz se utiliza como diccionario, el algoritmo SSC construye el modelo de representación disperso de la siguiente manera

$$\min_{\mathbf{C}} \|\mathbf{C}\|_1 \quad \text{s.t.} \quad \mathbf{X} = \mathbf{X}\mathbf{C}, \text{diag}(\mathbf{C}) = 0, \quad (4)$$

dónde la regularización de la norma  $\ell_1$  en esta formulación, sugiere que una representación dispersa de un punto de datos encuentra puntos del mismo subespacio. Utilizando la matriz de coeficientes dispersos obtenida  $\mathbf{C}$ , la matriz de afinidad  $\mathbf{A}$  es construida, la cual define los pesos en el borde entre los nodos de datos de la siguiente manera

$$\mathbf{A} = |\mathbf{C}| + |\mathbf{C}|^T. \quad (5)$$

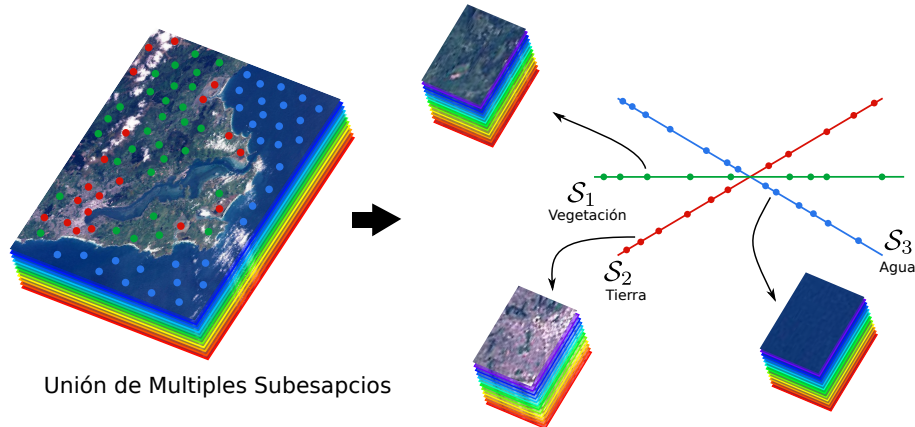
Finalmente, el resultado del agrupamiento se obtiene aplicando el agrupamiento espectral a la matriz Laplaciana  $\mathbf{L}$  inducida por la matriz de afinidad  $\mathbf{A}$ .

$$\mathbf{L} = \mathbf{I} - \mathbf{B}^{-1/2} \mathbf{A} \mathbf{B}^{-1/2}, \quad (6)$$

donde  $\mathbf{B} = \text{diag}\{b_i\}$  con  $b_i = \sum_{j=1}^n \mathbf{A}_{ij}$ .

Suponiendo que las firmas espectrales con características espectrales similares se encuentran en el mismo subespacio de baja dimensión, la teoría de agrupamiento del subespacio se puede utilizar para modelar el problema de clasificación de imágenes espectrales. La figura 3 representa una imagen espectral, cuyas firmas espectrales pertenecen a diferentes subespacios, es decir, puede verse como una unión de múltiples subespacios.

Este enfoque de agrupamiento de subespacios con restricción de escasez se puede resumir en el **Algoritmo 1**:



*Figura 3.* Clasificación de imágenes espectrales utilizando el enfoque de agrupamiento de subespacios (Hinojosa Montero, 2018).

---

**Algoritmo 1 : Agrupación de subespacios con restricción de escasez (SSC)**

---

**Parámetros:** Un conjunto de datos  $\{x_i\}_{i=1}^N$  provenientes de la unión de  $n$  subespacios lineales

$\{\mathcal{S}_i\}_{i=1}^n$

1. Resolver el problema de optimización Eq 21, mediante ADMM (ver Anexo A).
2. Normalizar las columnas de  $\mathbf{C}$  con  $c_i \leftarrow \frac{c_i}{\|c_i\|_\infty}$
3. Construir un grafo de similitud con  $N$  nodos que representan los datos. Establecer los pesos de los vértices mediante  $W = |C| + |C|^\top$
4. Aplicar agrupamiento espectral (Ng et al., 2002) sobre el grafo de similitud.

**Salida:** Asignación de los grupos de  $\mathbf{X}$

---

**1.2.1. Algoritmo escalable de agrupación de subespacios usando muestreo alea-**

**torio.** Por lo general, los algoritmos de agrupación de subespacios, descritos en la sección anterior son computacionalmente costosos debido a la alta dimensionalidad de los datos y además aplican el problema de optimización sobre todo el conjunto de datos. Esto causa que los algoritmos aborden el problema de datos a gran escala. Para disminuir el costo computacional algunos algoritmos

de agrupación de subespacios realizan muestreos sobre los datos para aplicar la optimización sobre esa muestra y después aplicar algunos métodos de codificación y clasificación sobre los puntos fuera de la muestra. Recientemente, se propuso un algoritmo escalable (Peng et al., 2013) (**Algoritmo 2**) el cual aborda los problemas a gran escala, el problema del muestreo, agrupación, codificación y clasificación. Para abordar el problema de gran escala y muestreo, seleccionan una muestra aleatoria de los datos y se calcula la relación entre ellos. Luego para agrupar los puntos fuera de la muestra calculan la relación entre ellos y los puntos dentro de la muestra agrupados mediante la proyección de puntos en subespacios y abordan la clasificación de los puntos mediante el cálculo del residuo con cada subespacio encontrado.

Para que la agrupación de este algoritmo se eficiente se basa en el siguiente postulado: Suponiendo que un conjunto de datos  $\mathbf{D}_i \in \mathbb{R}^{L \times n_i}$  se extrae del subespacio  $\mathcal{S}_i$ , donde  $i$  es un subespacio presente en el conjunto de datos  $\mathbf{D}$ . Se puede utilizar una pequeña porción de  $\mathbf{D}_i$ , denotada por  $\mathbf{X}_i \in \mathbb{R}^{L \times p_i}$ , para aprender la estructura de  $\mathcal{S}_i$ , donde el rango( $\mathbf{D}_i$ )=rango( $\mathbf{X}_i$ ), rango( $\mathbf{X}_i$ )  $\leq p_i \ll n_i$ , y  $\mathcal{S}_i$  es un espacio métrico compacto.

Este postulado implica dos hechos. Primero, cada punto de datos podría escribirse como una combinación lineal de algunas bases (es decir, suposición de escasez). Segundo, requiere que  $\mathbf{X}_i$  y  $\mathbf{D}_i$  sean independientes y distribuidos de manera idéntica (i.i.d., por sus siglas en inglés) para que los datos fuera de la muestra puedan ser representados por  $\mathbf{X}_i$ . Este postulado es muy general y en el que se basan la mayoría de los trabajos de minería de datos y aprendizaje automático.

En la práctica, la suposición de escasez se satisface fácilmente para datos de alta dimensión, como las imágenes espectrales. Para satisfacer el supuesto de i.i.d., necesitamos encontrar

los puntos representativos  $\mathbf{X}_i \in \mathbb{R}^{L \times p}$  desde  $\mathbf{D}_i \in \mathbb{R}^{L \times n}$  para que los datos fuera de la muestra  $\mathbf{Y}_i \in \mathbb{R}^{L \times (n-p)}$  sean ubicados en los subespacios abarcados por  $\mathbf{X}$ . Para tener en cuenta, algunas técnicas de muestreo, como el método de selección propuesto en (Halko et al., 2011) puede ser usado. Sin embargo, estos métodos de muestreo son ineficientes y no se pueden aplicar a problemas de gran escala. En este algoritmo se adopta un método de muestreo aleatorio uniforme en el que el costo de tiempo es solo de  $O(1)$ .

Este método escalable de agrupamiento de subespacios se puede resumir en el siguiente algoritmo:

---

**Algoritmo 2 : Agrupación escalable de subespacios utilizando muestreo aleatorio**

---

**Parámetros:** Un conjunto de datos  $\mathbf{D} \in \mathbb{R}^{L \times n}$ , el número deseado de grupos a encontrar  $k$  y el parámetro de regresión rígida  $\gamma$

1. Se selecciona aleatoriamente  $p$  puntos aleatorios de  $\mathbf{D}$  como datos fuera de la muestra  $\mathbf{X} = (\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_p)$ . Las muestras restantes se utilizan como datos dentro de la muestra  $\mathbf{Y} = (\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_{n-p})$ .
2. Se aplica un algoritmo de agrupación de subespacios, como SSC (**Algoritmo 1**), sobre el conjunto de datos  $\mathbf{X}$  para obtener la relación de grupos presentes en  $\mathbf{X}$ .
3. Se proyecta cada punto de datos fuera de la muestra  $\mathbf{y}_i$  en la unión de los subespacios que se

encuentran en  $\mathbf{X}$  a través de la solución de:

$$c_i^* = (\mathbf{X}^T \mathbf{X} + \gamma \mathbf{I})^{-1} \mathbf{X}^T \mathbf{y}_i, \quad (7)$$

donde  $I$  es la matriz identidad y  $\gamma$  es el parámetro de regresión rígida.

4. Se calculan los residuos de  $\mathbf{y}_i$  sobre el  $j$ -ésimo subespacio con:

$$r_j(\mathbf{y}_i) = \|\mathbf{y}_i - \mathbf{X} \delta_j(c_i^*)\|_2, \quad (8)$$

o los residuos normalizados de  $\mathbf{y}_i$  sobre todos los subespacios a través de la resolución

$$r_j(\mathbf{y}_i) = \frac{\|\mathbf{y}_i - \mathbf{X} \delta_j(c_i^*)\|_2}{\|\delta_j(c_i^*)\|_2}, \quad (9)$$

donde  $\delta_j$  es un parámetro de tolerancia al error.

5. Se asigna  $\mathbf{y}_i$  al subespacio que tiene el mínimo residuo dado por

$$f(\mathbf{y}_i) = \underset{j}{\operatorname{argmin}} \{r_j(\mathbf{y}_i)\}. \quad (10)$$

**Salida:** Asignación de los grupos de  $\mathbf{D}$

---

### 1.3. Segmentación de imágenes

En el campo de visión por computador, la segmentación de imágenes es el proceso de partición de una imagen digital en varios segmentos (conjuntos de píxeles). El objetivo de la segmentación es simplificar y/o cambiar la representación de una imagen en una imagen más significativa y más fácil de analizar (Shapiro and George, 2002). La segmentación de imágenes se suele utilizar para ubicar objetos y límites (líneas, curvas, etc.) en las imágenes. El resultado de la segmentación de una imagen es un conjunto de segmentos que cubren colectivamente toda la imagen, o un conjunto de contornos extraídos de la imagen. Cada uno de los píxeles en una región es similar con respecto a alguna característica o propiedad calculada, como el color, la intensidad o la textura. Las regiones adyacentes son significativamente diferentes con respecto a las mismas características (Shapiro and George, 2002).

**1.3.1. Detección de bordes y umbralización.** La detección de bordes incluye una variedad de métodos matemáticos que apuntan a identificar puntos en una imagen digital en los que el brillo de la imagen cambia bruscamente o, más formalmente, tiene discontinuidades. Las discontinuidades son detectadas usando derivadas de primer y segundo orden, en el caso de derivadas de primer orden se utiliza el operador gradiente, mientras que en derivadas de segundo orden se utiliza el operador Laplaciano (Palomino and Concha, 2009). Los puntos en los que el brillo de la imagen cambia bruscamente se organizan típicamente en un conjunto de segmentos de líneas o curvas denominados bordes. El mismo problema de encontrar discontinuidades en señales unidimensionales se conoce como detección de pasos y el problema de encontrar discontinuidades de

señales en el tiempo se conoce como detección de cambios. La detección de bordes es una herramienta fundamental en el procesamiento de imágenes, visión artificial y visión por computadora, particularmente en las áreas de detección y extracción de características (Umbaugh, 2010).

Matemáticamente la manera de encontrar discontinuidades en una función es por medio de la derivada, a continuación se describe la forma de calcular las discontinuidades por medio del operador gradiente. El gradiente de una imagen es una función  $f(x,y)$  que se define como el vector bidimensional presentado en la siguiente ecuación:

$$\mathbf{G}(F(x,y)) = \begin{bmatrix} G_x \\ G_y \end{bmatrix} = \begin{bmatrix} dF/dx \\ dF/dy \end{bmatrix} \quad (11)$$

El operador gradiente  $\mathbf{G}$  alcanza su máximo valor en la dirección en que la variación es máxima, por lo tanto puede definirse como:

$$\mathbf{G}(F(x,y)) = \sqrt{[G_x(x,y)]^2 + [G_y(x,y)]^2} \quad (12)$$

A partir de las ecuaciones 11 y 12, se deduce que el cálculo del gradiente de una imagen se basa en la obtención de las derivadas parciales  $dF/dx$  y  $dF/dy$  de manera discreta sobre cada píxel. Las derivadas de primer orden permiten encontrar lugares en una imagen en donde la intensidad que cambia rápidamente es mayor en magnitud que un umbral especificado (Palomino and Concha, 2009). Existen varios estimadores derivativos como los de Sobel, Prewit y el Laplaciano.

- El detector de bordes de Sobel, analiza los bordes en dirección horizontal como vertical, y

combina esta información en una sola medida (MUÑOZ et al., 2001). A continuación se muestran las máscaras empleadas para la detección de bordes en dos direcciones (González, 1996) (Trucco and Verri, 1998):

$$G_x = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}, G_y = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \quad (13)$$

Estas máscaras se convolucionan con la imagen. En cada píxel ahora tenemos dos medidas:  $r_1$  y  $r_2$ , que corresponde a las máscaras de fila y columna respectivamente, con estas medidas se halla la magnitud y la dirección del borde, que están definidos así:

$$R = \sqrt{r_1^2 + r_2^2} \quad D = \tan^{-1} \left[ \frac{r_1}{r_2} \right] \quad (14)$$

- El operador de Prewitt es similar al operador de Sobel, pero con diferentes coeficientes en las máscaras de convolución, como se muestra a continuación (González, 1996):

$$G_x = \begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix} \quad G_y = \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix} \quad (15)$$

La magnitud y la dirección del borde se definen de la misma forma que para el operador de

Sobel descritos en la ecuación 14.

- El operador Laplaciano marca la presencia de un borde cuando la salida del operador realiza una transición por cero. En principio, las localizaciones de los cruces por cero pueden ser estimadas a nivel de subpíxeles usando interpolación lineal; pero los resultados pueden ser inexactos. El Laplaciano de una función bidimensional  $f(x,y)$  es una derivada de segundo orden y está definida por:

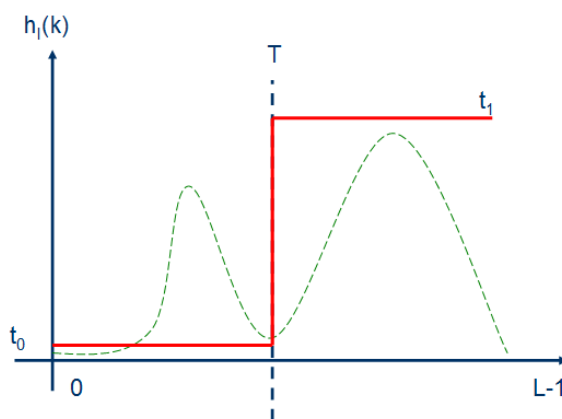
$$\nabla^2 f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} \quad (16)$$

Se pueden utilizar aproximaciones discretas del operador Laplaciano para ser utilizadas como máscaras de convolución, en la siguiente ecuación se presentan tres aproximaciones discretas de máscaras de convolución (Haralick and Shapiro, 1992):

$$\begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 0 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} 1 & 1 & 1 \\ 1 & -8 & 1 \\ 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} -1 & 2 & -1 \\ 2 & -4 & 2 \\ -1 & 2 & -1 \end{bmatrix} \quad (17)$$

- La umbralización es uno de los métodos ampliamente utilizados para la segmentación de imágenes. Es útil para discriminar el primer plano del fondo. Al seleccionar un valor umbral  $T$  adecuado, la imagen de nivel de gris se puede convertir en imagen binaria. La imagen binaria debe contener toda la información esencial sobre la posición y la forma de los objetos

de interés (primer plano). La ventaja de obtener primero una imagen binaria es que reduce la complejidad de los datos y simplifica el proceso de reconocimiento y clasificación. La forma más común de convertir una imagen de nivel de gris en una imagen binaria es seleccionar un único valor de umbral ( $T$ ). Entonces, todos los valores de nivel de gris debajo de este  $T$  se clasificarán como negros (0), y los de arriba de  $T$  serán blancos (1). El problema de segmentación se convierte en seleccionar el valor apropiado para el umbral  $T$  ver figura 4. Un método frecuente utilizado para seleccionar  $T$  es analizar los histogramas del tipo de imágenes que se quieren segmentar. El caso ideal es cuando el histograma presenta solo dos modos dominantes y un valle claro (bimodal). En este caso, el valor de  $T$  se selecciona como el punto del valle entre los dos modos. En aplicaciones reales, los histogramas son más complejos, con muchos picos y valles no claros (Al-Amri et al., 2010).



*Figura 4.* Histograma bimodal.  $T$  es el valor de intensidad umbral y  $t_0$  y  $t_1$  son los dos posibles niveles de gris de los píxeles de la imagen umbralizada (Osorio et al., 2011).

**1.3.2. Super píxeles.** El concepto de superpíxel fue introducido por primera vez por Xiaofeng Ren y Jitendra Malik en 2003 (Ren and Malik, 2003). A través de la definición de las

regiones perceptualmente uniformes utilizando el algoritmo de corte normalizado (NCuts). Los superpíxeles agrupan los píxeles en regiones simétricas de significado perceptual que se pueden usar para reemplazar la estructura rígida de la cuadrícula de píxeles en las imágenes. De esta manera, se puede eliminar la redundancia en gran magnitud. Además, para las aplicaciones subsiguientes, es más conveniente y efectivo calcular las características de la imagen basadas en regiones que en píxeles. En general, los algoritmos de segmentación de superpíxeles se emplean generalmente como pasos de procesamiento de muchas tareas de visión artificial para mejorar sus actuaciones, como la segmentación de imágenes (Peng et al., 2011), reconocimiento de objetos (Li et al., 2012).

El algoritmo de agrupamiento iterativo lineal simple (Achanta et al., 2012) (**SLIC**, por sus siglas en inglés) es el más utilizado hoy en día, ya que este trabaja en un espacio 5D, con dos dimensiones en el espacio  $(x, y)$  y sus otras tres dimensiones son las tres componentes del espacio de color RGB. Dado el tamaño deseado de los superpíxeles  $N_{seg}$ , donde el tamaño aproximado de cada superpíxel es  $n/N_{seg}$ , el primer paso de SLIC es definir un centro de clúster en cada intervalo de cuadrícula  $Sr = \sqrt{n/N_{seg}}$ . El algoritmo asume que los píxeles asociados con un clúster se encuentran en una  $2Sr \times 2Sr$  zona alrededor del centro de superpíxeles en el plano  $(x, y)$ . Por lo tanto, esto se convierte en el área de búsqueda de los píxeles cercanos a cada centro del clúster. Para evitar mantener el centro en el borde de un objeto, se transfiere a la posición de gradiente más baja en un vecindario de  $3 \times 3$ .

En el siguiente paso, para cada centro de clúster, SLIC asigna los píxeles que mejor coinci-

den del área de búsqueda según la medida de distancia,

$$D_c = \sqrt{(R_j - R_{j'})^2 + (G_j - G_{j'})^2 + (B_j - B_{j'})^2} \quad (18)$$

$$D_p = \sqrt{(x_j - x_{j'})^2 + (y_j - y_{j'})^2} \quad (19)$$

$$D_t = D_c + \frac{m}{S} D_p \quad (20)$$

donde  $R_j, G_j, B_j$  corresponde al  $j$ -ésimo píxel, que normalmente se elige como  $m = 10$ . La complejidad del algoritmo SLIC es lineal  $O(n)$ , ya que necesita calcular distancias desde cualquier punto hasta no más de ocho centros de agrupación y el número de iteraciones es constante (Achanta et al., 2012). Observe que la región crece como máximo 2 veces el radio del clúster con el algoritmo SLIC.

## 2. Método Propuesto

En general los algoritmos de agrupamiento subespacial asumen que los datos provienen de la unión de múltiples subespacios de baja dimensión y que se desconoce a qué subespacio pertenece cada punto. Los algoritmos de agrupación subespacial del estado del arte como SSC, busca representar cada punto como la combinación lineal de los otros puntos dentro del conjunto de datos y luego imponen una restricción de esparcidad sobre el conjunto de soluciones. De esta forma, los puntos que permiten una mejor representación se asume que pertenecen al mismo subespacio. Sin embargo estos métodos no son escalables y al aumentar el número de puntos, el rendimiento es afectado notablemente, esto hace que sea difícil la utilización de estas técnicas en implementaciones reales como en HSI. Adicionalmente, el modelo de SSC no tiene en cuenta la información espectral presente en una HSI para tomar una mejor decisión de clasificación. Para abordar el problema de gran escala, recientemente han desarrollado una idea la cual consiste en realizar un muestreo aleatorio de los datos y aplicar en ellos algoritmos de agrupamiento de subespacios para obtener su respectiva asignación de grupos (Peng et al., 2013). Luego, sobre los puntos restantes aplicar a cada uno, una proyección sobre cada subespacio y asignarlo al subespacio sobre el cual el residuo sea más pequeño, ver **Algoritmo 2**. Sin embargo, este algoritmo no tiene en cuenta la información espacial de las HSI y no es preciso al momento de agrupar puntos donde los píxeles presentan cambios de intensidad. Por lo tanto este método sería más efectivo si el muestreo se hiciera ignorando aquellos puntos los cuales están cercanos a los cambios de intensidad en la imagen, y dejándolos para la asignación de grupos mediante la proyección y el residuo.

Teniendo en cuenta lo anterior este trabajo de investigación propone un nuevo enfoque de agrupamiento de subespacios en HSI considerando la información espacial de estas, para realizar un muestreo restringido por la similitud de los puntos aplicando técnicas de segmentación de imágenes, tales como la detección de bordes y superpíxeles, ver figura 5. Esto con el fin de mejorar considerablemente la precisión del agrupamiento y los tiempos de ejecución.

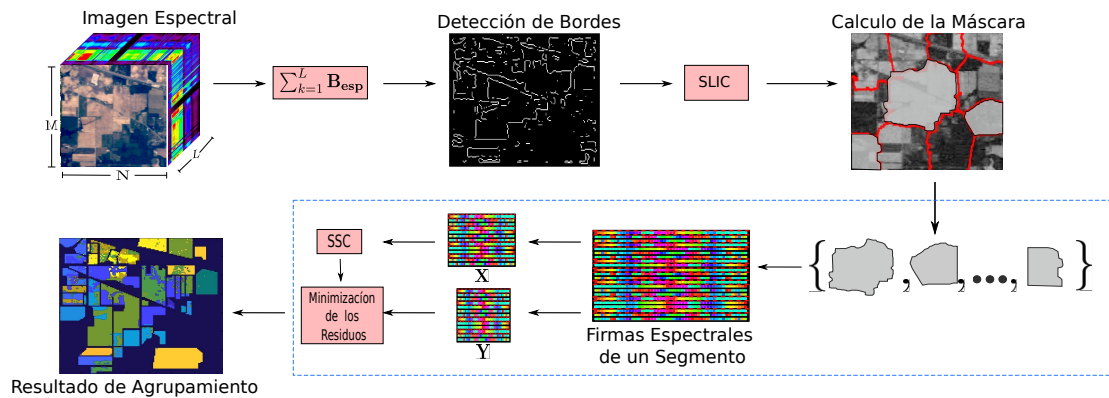


Figura 5. Método de agrupación de subespacios propuesto.

## 2.1. Algoritmo de agrupación de subespacios con restricción de escasez (SSC)

El problema de optimización mostrado en 4 es adecuado solo para condiciones ideales, es decir, cuando la adquisición de la imagen no incluye algún tipo de ruido. Para considerar el ruido, Elhamifar y Vidal (Elhamifar and Vidal, 2009, 2013) proponen un problema de optimización para SSC con una teoría de recuperación bien fundada para subespacios independientes y subespacios disjuntos, en la cual considera el ruido presente en la escena resolviendo el siguiente problema de

optimización:

$$\begin{aligned} \min_{\mathbf{C}, \mathbf{E}, \mathbf{Z}} \|\mathbf{C}\|_1 + \frac{\lambda_Z}{2} \|\mathbf{Z}\|_F^2 \\ \text{s.t. } \mathbf{X} = \mathbf{XC} + \mathbf{Z}, \text{diag}(\mathbf{C}) = 0, \end{aligned} \quad (21)$$

donde  $\mathbf{C} \in \mathbb{R}^{n \times n}$  es la representación escasa del conjunto de datos,  $\mathbf{X} \in \mathbb{R}^{L \times n}$  son las firmas espectrales de la HSI,  $\mathbf{Z}$  denota los errores de reconstrucción para la capacidad de representación limitada, y  $\lambda_Z$  corresponden a un parámetro de regularización.

Una vez obtenida la matriz de representación escasa  $\mathbf{C}$ , SSC construye el grafo de similitud resolviendo  $\mathbf{A} = |\mathbf{C}|^T + |\mathbf{C}|$  y aplica agrupamiento espectral (Ng et al., 2002) sobre este, utilizando el método de dirección alterna de multiplicadores (ADMM, por sus siglas en inglés), ver apéndice A.

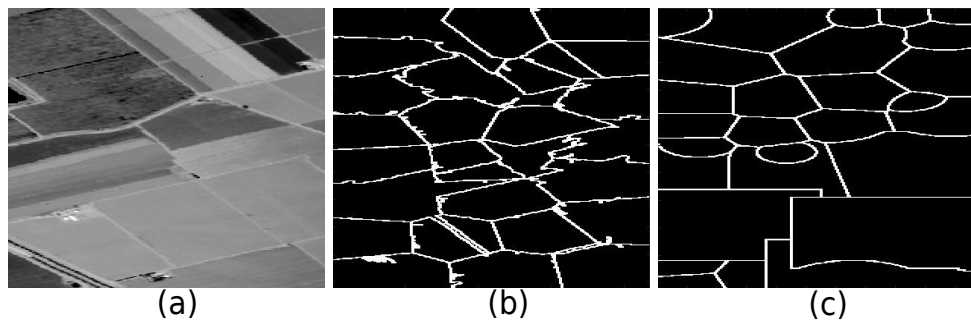
## 2.2. Detección de bordes y superpíxeles

Comúnmente los métodos de agrupación de subespacios, como SSC, pueden ser fácilmente aplicados sobre HSI, considerando solo la información que brindan las firmas espectrales. Sin embargo, estos algoritmos no tienen en cuenta la información espacial de la HSI a la hora de decidir a qué clase pertenece punto. En este trabajo se propone un método que aborda la agrupación de subespacios, considerando la información espacial y espectral presentes en la imagen. Para considerar la información espacial presente en la HSI se aplicará detección de bordes y superpíxeles.

La detección de bordes en imágenes en escala de grises se ha estudiado a fondo y está bien establecida (Trucco and Verri, 1998). Sin embargo, para las imágenes multicanal, como las HSI este tema está menos desarrollado, ya que, incluso definir bordes para estas imágenes es un desafío

debido a la alta dimensionalidad de los datos. Por lo tanto, en este trabajo se utiliza un enfoque práctico, el cual consiste en sumar todas las bandas espectrales de una imagen hiperespectral y obtener una imagen de dos dimensiones  $\mathbf{H}^* \in \mathbb{R}^{M \times N}$ , a la cual se aplicara el algoritmo de detección de bordes Sobel 1.3.1 eligiendo el umbral de forma heurística, en función de la imagen de entrada mediante la función `edge` en MATLAB.

Una vez obtenida la imagen con los bordes  $\mathbf{B} \in \{0, 1\}^{M \times N}$  se aplica el algoritmo de superpíxeles SLIC, con el fin de obtener una mejor estructura de los subespacios presentes en la escena ver figura 6(b), y así obtener una mejor clasificación.



*Figura 6.* (a) Representación en escala de grises de una HSI. (b) Resultado de la detección de bordes mediante la suma de todas las bandas espectrales de la HSI y luego aplicar superpíxeles. (c) Suma de todas las bandas de la HSI y calcular superpíxeles.

### 2.3. Muestreo restringido por similitud

Teniendo en cuenta que los píxeles dentro de un superpíxel guardan una alta correlación espacial, donde dicha correlación es más alta entre los puntos cercanos al centroide y mucho más baja en los bordes o fronteras del superpíxel. Este trabajo busca desarrollar un enfoque eficiente, similar a (Peng et al., 2013), (ver **Algoritmo 2**), donde la separación de los puntos de datos se realizará dentro de cada segmento y dicha separación tendrá como criterio la similitud espacial de

los píxeles. El mapa de superpíxeles puede ser visto como una imagen  $\mathbf{S}_p \in \mathbb{R}^{M \times N}$ , los cuales son calculados sobre la imagen  $\mathbf{B}$ . Como la imagen  $\mathbf{S}_p$  esta formada por la unión de todos los segmentos  $\mathbf{S}_p = [\mathbf{s}_{p_1}, \mathbf{s}_{p_2}, \dots, \mathbf{s}_{p_{N_s}}]$ , donde estos segmentos presentan fronteras, se calculará una máscara la cual consiste en tomar, alrededor de estas fronteras, un píxel adyacente en la dirección horizontal, vertical o diagonal, con el fin de tener una separación más adecuada de los puntos pertenecientes a un segmento. Los puntos que no hagan parte de la máscara se denotarán como puntos fuera de la máscara y a los puntos que sí, se les llamará puntos dentro de la máscara, ver figura 7.

Sobre los datos pertenecientes a cada segmento se aplica el **Algoritmo 2**, solo con la pequeña modificación de que los puntos dentro y fuera de la muestra no se seleccionan aleatoriamente. Los puntos dentro la muestra  $\mathbf{X}$  serán aquellos puntos los cuales se encuentren por fuera de la máscara y los puntos fuera de la muestra  $\mathbf{Y}$  serán los puntos dentro de la máscara.

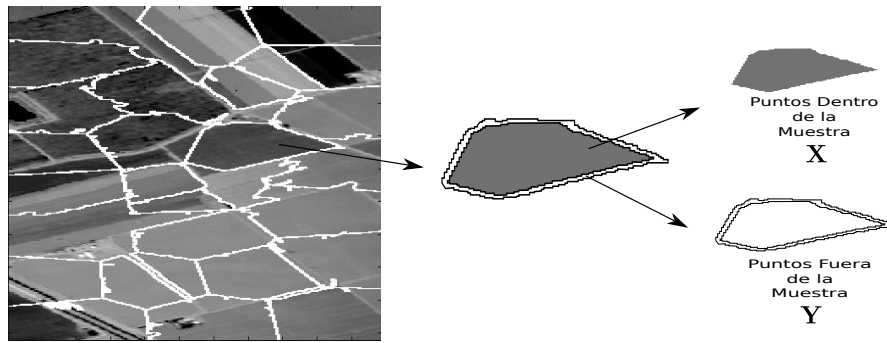


Figura 7. Selección de puntos dentro y fuera de la muestra.

En la práctica pueden existir segmentos en los cuales no se conozca la cantidad de grupos que hay en él. Debido a que uno de los parámetros del **Algoritmo 2** es  $k_i$  el cual es la cantidad de grupos presentes en un segmento  $\mathbf{s}_i$ , no se puede aplicar este algoritmo a los puntos presentes. Para poder abordar este problema se propone utilizar una matriz de etiquetas  $\mathbf{Ac} \in \mathbb{R}^{M \times N}$  la cual

es inicializada en cero y cambiará su valor por 1 en los puntos pertenecientes al segmento que no se conoce el valor de  $k_i$ . Para identificar si esta matriz de ceros es modificada y guarda puntos los cuales no pudieron ser agrupados, se utilizará una variable bandera booleana que cambiará su valor inicial si en la matriz  $\mathbf{Ac}$  existe el valor de 1. Una vez se halla agrupado algún registro con el resto de puntos se verifica si la variable booleana ha cambiado su valor inicial, y si es así se hace una proyección de cada punto no agrupado sobre los puntos agrupados y serán asignados al subespacio más cercano mediante el cálculo de residuos, en otras palabras se aplicaría nuevamente el **Algoritmo 2** a partir del paso 3, pero no se tomarán todos los puntos agrupados, solo un porcentaje de ellos serán seleccionados aleatoriamente. Este porcentaje, que denotaremos  $\phi$ , será un parámetro a discutir en este trabajo.

Otra manera de abordar el problema de no saber la cantidad de subespacios presentes en un segmento  $k_i$  es utilizando un método basado en la descomposición de Cholesky (Von Luxburg, 2007) sobre la matriz Laplaciana  $\mathbf{L}$ , del grafo de similitud. El valor de  $k_i$  se estima mediante:

$$k_i = \underset{i \in n-1}{\operatorname{argmax}}(\lambda_{i+1} - \lambda_i), \quad (22)$$

donde  $\lambda_1 \leq \lambda_2 \leq \dots \lambda_{n-1} \leq \lambda_n$  son los auto valores normalizados de la matriz Laplaciana  $\mathbf{L}$ .

El siguiente algoritmo resume el método propuesto de agrupación de subespacios sobre imágenes espectrales utilizando muestreo restringido por similitud.

---

**Algoritmo 3 : Algoritmo de agrupación de subespacios con muestreo restringido por simili-**

tud (SRSSSC)

---

**Parámetros:** Imagen hiperespectral  $\mathbf{H} \in \mathbb{R}^{M \times N \times L}$ , la cantidad de superpíxeles deseados a calcular  $\mathbf{N}_{\text{seg}}$ , la cantidad de grupos presentes en cada segmento  $\mathbf{K}$  y el porcentaje de los puntos no clasificados a proyectar  $\phi$

1.  $\mathbf{H}^* \leftarrow \sum_{i=1}^L \mathbf{H}(:, :, i)$  // Obtención de una imagen 2D a partir de la suma de bandas de la HSI

2.  $ubr \leftarrow Sobel(\mathbf{H}^*)$

3.  $\mathbf{B} \leftarrow sobel(\mathbf{H}^*, ubr)$

4.  $\mathbf{S}_p \leftarrow \text{SLIC}(\mathbf{B}, \mathbf{N}_{\text{seg}})$  // Calculo de superpíxeles

5. Encontrar la máscara de  $\mathbf{S}_p$

6. **for**  $t \leftarrow 1$  **hasta**  $\mathbf{N}_{\text{seg}}$  **Realizar**

**if**  $k_t \neq 0$

$\mathbf{G}^{(t)} \leftarrow \text{Algoritmo 2 sobre } \mathbf{s}_t$

**else**

$\mathbf{Ac}^{(t)} \leftarrow \mathbf{Ac}^{(t)} \cup \mathbf{Ac}^{(t-1)}$

$aux = true$

**Finalizar for**

**if**  $aux == true$

**for**  $tt \leftarrow 1$  **hasta**  $\max(\mathbf{G})$  **Realizar**

$ind \leftarrow (G == tt)$  // Selección de los índices que contiene un grupo  $s_i$

$L_G \leftarrow \text{lenght}(ind)$  // Tamaño del vector

$indr \leftarrow \text{permutacion}(ind)$  // Desordenar el vector  $ind$

$indr = indr[1 : \phi L_g]$  // Seleccionar un porcentaje del vector

$\tilde{G} \leftarrow \tilde{G} \cup G(indr)$

**Finalizar for**

$c_i^* \leftarrow (\tilde{\mathbf{G}}^T \tilde{\mathbf{G}})^{-1} \tilde{\mathbf{G}}^T \mathbf{A} \mathbf{c}_i$  //Proyección sobre los subespacios

$r_j(\mathbf{y}_i) \leftarrow \frac{\|\mathbf{A} \mathbf{c}_i - \tilde{\mathbf{G}} \delta_j(c_i^*)\|_2}{\|\delta_j(c_i^*)\|_2}$  //Cálculo de los residuos

$f(\mathbf{y}_i) \leftarrow \text{argmin}_j \{r_j(\mathbf{y}_i)\}$ . // Asignación al subespacio con residuo mínimo

**Salida:** La asignación de grupos de  $\mathbf{H}$ .

---

## 2.4. Análisis de Complejidad Computacional

La complejidad computacional del método propuesto es  $N_{seg}t_1L\psi^3 + N_{seg}t_2\psi k_i^2 + N_{seg}n\psi^2 + 2n$ . Donde  $\psi$  es la cantidad de muestras seleccionadas de los puntos pertenecientes a un segmento. En otras palabras  $\psi$  es la cantidad de columnas de  $\mathbf{X}$ . En el siguiente cuadro 1 se muestra a que corresponde cada termino:

Termino	Descripción
$N_{seg}t_1L\psi^3$	Construcción del grafo de similitud
$N_{seg}t_2\psi k_i^2$	Agrupamiento espectral (Ng et al., 2002)
$N_{seg}n\psi^2$	Proyección de los puntos dentro de la muestra agrupados
$n$	SLIC
$n$	Detección de bordes

Tabla 1

*Complejidad temporal y espacial del algoritmo propuesto, donde  $t_1$  y  $t_2$  son las iteraciones necesarias para resolver el problema de optimización mostrado en 21.*

Como se muestra en el siguiente cuadro 2 la complejidad computacional del método propuesto en este trabajo es menor a los dos algoritmos de agrupamiento de subespacios mostrados en el transcurso de este trabajo ( **Algoritmo 1** (SSC) y **Algoritmo 2** (SSSC) ), esto sujeto a que

$$\psi < \frac{n}{N_{seg}} \text{ y } \psi \ll n.$$

Algoritmo	Complejidad Temporal	Complejidad Espacial
SSC	$t_1 Ln^3 + t_2 nk^2$	$Ln^2$
SSSC	$t_1 Lp^3 + t_2 pk^2 + np^2$	$Lp^2$
SRSSSC	$N_{seg}t_1 L\psi^3 + N_{seg}t_2 \psi k_i^2 + N_{seg}n\psi^2 + 2n$	$L\psi^2$

Tabla 2

Complejidad computacional de SSC, SSSC, y el algoritmo propuesto en este trabajo.  $t_1$  corresponde al número de iteraciones necesarias para la convergencia de la norma  $\ell_1$  en el problema de optimización 21.  $t_2$ , son las iteraciones para el algoritmo k-means.

### 3. Simulaciones y resultados

El enfoque de agrupación subespacial en imágenes espectrales utilizando muestreo restringido por similitud se probó en tres conjuntos de datos hiperespectrales reales, con diferentes entornos de imágenes. Estas imágenes fueron adquiridas por el espectrómetro de imágenes Visibles/Infrarrojas aerotransportadas (**AVIRIS**, por sus siglas en inglés) y el espectrómetro de imágenes del sistema de óptica reflectante (**ROSIS**, por sus siglas en inglés). Específicamente, se utilizaron el conjunto de datos de Indian Pines, el conjunto de datos de Salinas y el conjunto de datos de la universidad de Pavia.

- **Indian Pines:** Esta HSI fue adquirida, por el sensor AVIRIS del sitio de prueba Northwestern Indian Pines en junio de 1992 (Zhang et al., 2016). Las dimensiones de esta imagen es de  $145 \times 145$ . Se eliminaron de la imagen original 20 bandas espectrales de las 220 bandas originales, que contenían absorción de agua y ruido (104 – 108, 150 – 163 y 220), dejando así 200 bandas para las simulaciones (Zhang et al., 2016). Las cuales contiene información de 16 clases: alfalfa, maíz pequeño, maíz mediano, maíz grande, Pastizales, xantorroeóideas, segado de pastos, heno en hileras, avena, soja pequeña, soja mediana, soja limpia, trigo, bosque. Estas clases se muestran en la imagen de etiquetas de las clases presentes 8(b) y se muestra una imagen de falso color 8(a).
- **Salinas:** Este segundo conjunto de datos contiene 224 bandas espectrales y fue adquirido por el sensor AVIRIS sobre el Valle de Salinas, CA, USA. La resolución de esta imagen

es de  $512 \times 217$ . Al igual que el primer conjunto de datos, se han removido un total de 20 bandas espectrales que contenían absorción de agua (108 – 112, 154 – 167, y 224). Esta imagen cuenta con 16 clases las cuales son: brócoli-hierbas verdes 1, brócoli-hierbas verdes 2, barbecho, arado áspero en barbecho, barbecho liso, rastrojo, apio, uvas, viñedo, viñedo, lechuga romana de 4 semanas, lechuga romana de 5 semanas, lechuga romana de 6 semanas, lechuga romana de 7 semanas, viñedo sin entrenar, viñedo enrejado verticalmente. Estas clases se muestran en la imagen de etiquetas de las clases presentes 9(b) y se muestra una imagen de falso color 9(a).

- **Universidad de Pavia:** Esta imagen fue adquirida por el sensor ROSIS durante un vuelo sobre Pavía, norte de Italia. El tamaño de esta imagen es de  $610 \times 340$ , con 103 bandas espectrales. Las cuales contienen información de 8 clases: asfalto, prado, grava, árboles, hojas de metal pintadas, suelo desnudo, betún, ladrillos autobloqueantes, oscuridad. Estas clases se muestran en la imagen de etiquetas de las clases presentes 10(b) y se muestra una imagen de falso color 10(a).

La clasificación no supervisada de estas imágenes es, en general, una tarea difícil debido a que las firmas espectrales de las clases de cobertura del suelo en estas áreas son muy similares y algunas de las curvas espectrales son mixtas. Y esto se puede evidenciar en las figuras 8(c), 9(c), 10(c).

En los experimentos, el número de grupos  $k_i$  presente en un segmento se estableció como

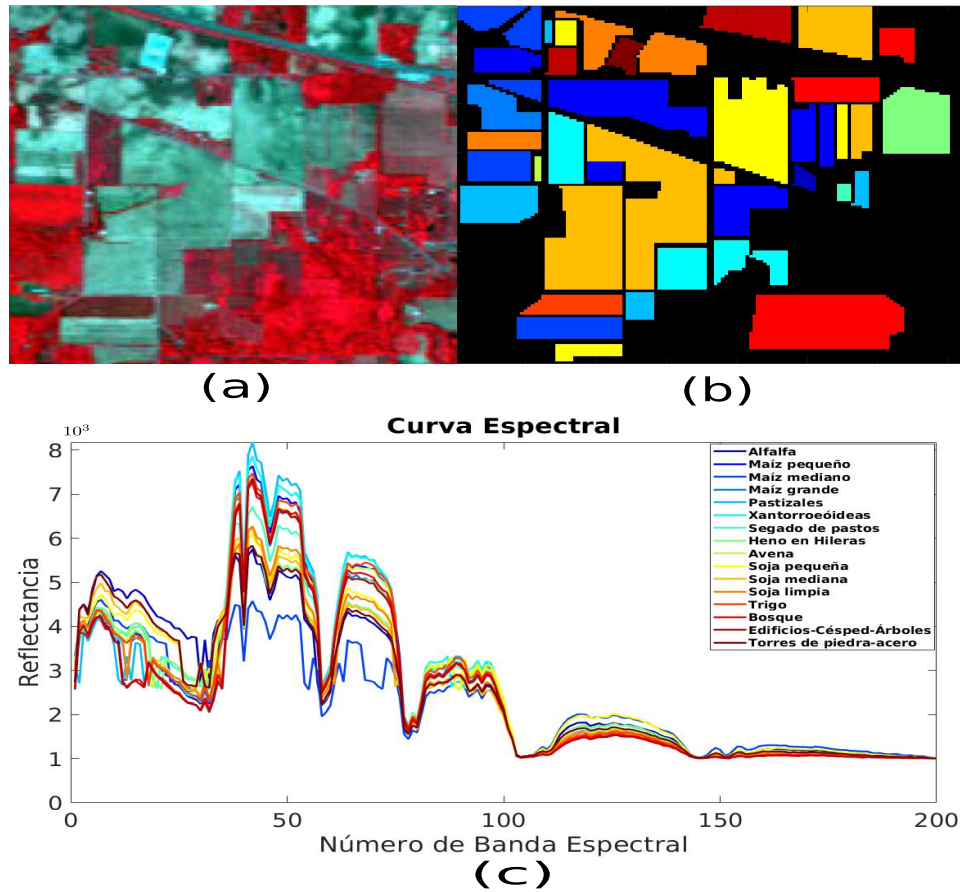


Figura 8. Indian Pines. (a) Imagen en falso color. (b) Imagen de etiquetas de las clases presentes. (c) Curvas espectrales de todas las clases.

una entrada manual para el algoritmo de agrupación de subespacios, el parámetro de regresión rígida  $\gamma$  para el **Algoritmo 2** se mantuvo constante con un valor de  $10^{-6}$ .

Las métricas utilizadas para evaluar la precisión de agrupamiento sobre estas imágenes fueron: Precisión general (OA), precisión promedio (AA), kappa y información mutua normalizada (NMI). La definición de estas métricas se muestran en la sección 2.

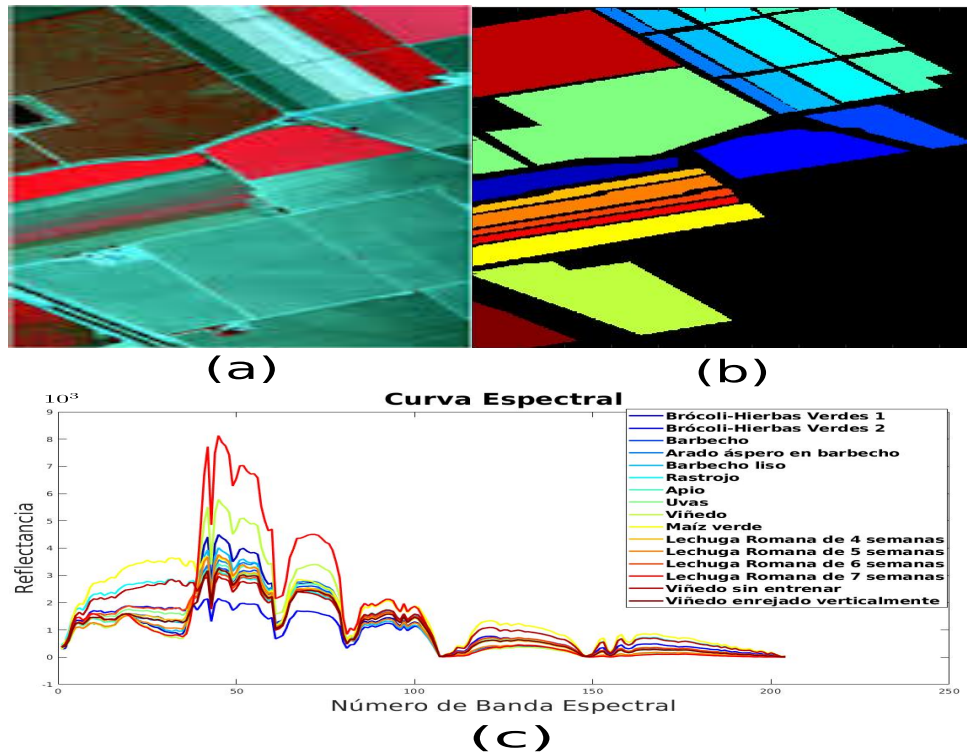


Figura 9. Salinas. (a) Imagen en falso color. (b) Imagen de etiquetas de las clases presentes. (c) Curvas espectrales de todas las clases.

### 3.1. Análisis del parámetro $\phi$

Debido a que la resolución espacial de cada imagen utilizada es diferente, se presentarán los resultados de agrupamiento en función de  $\rho$ , el cual representa la relación entre la cantidad de píxeles de la imagen y el número de segmentos. En otras palabras,  $\rho$  se puede interpretar como la cantidad de píxeles presentes en un superpíxel asumiendo que la cantidad de píxeles en los segmentos es uniforme, teniendo en cuenta que el algoritmo SLIC calcula todos los segmentos con

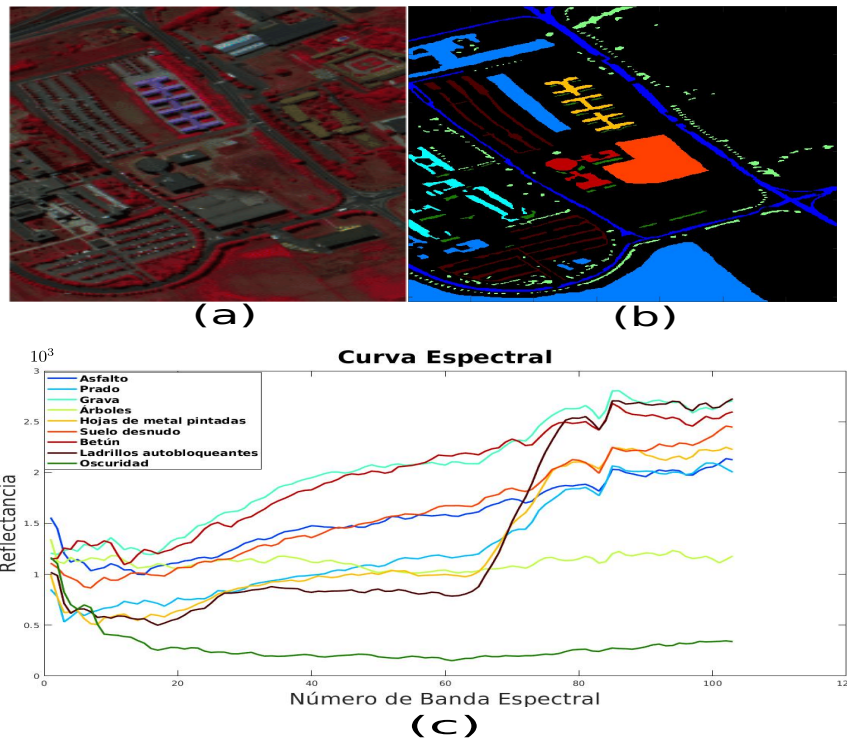


Figura 10. Universidad de Pavia. (a) Imagen en falso color. (b) Imagen de etiquetas de las clases presentes. (c) Curvas espectrales de todas las clases.

la misma cantidad de píxeles :

$$\rho = \frac{n}{N_{seg}}. \quad (23)$$

Todos los resultados de agrupamiento se presentarán en función de  $1/\rho$ , esto con fines de una mejor visualización de los resultados.

A continuación se muestra la influencia del parámetro  $\phi$ , el cual representa el porcentaje de datos seleccionados aleatoriamente de los puntos ya agrupados, para ser utilizados en la asignación de grupos a los puntos que no se pudieron agrupar debido a que se desconocía la cantidad de grupos

presentes  $k_i$  en algunos segmentos. La influencia que tiene este porcentaje  $\phi$  sobre el agrupamiento en la imagen Salinas se muestra a continuación:

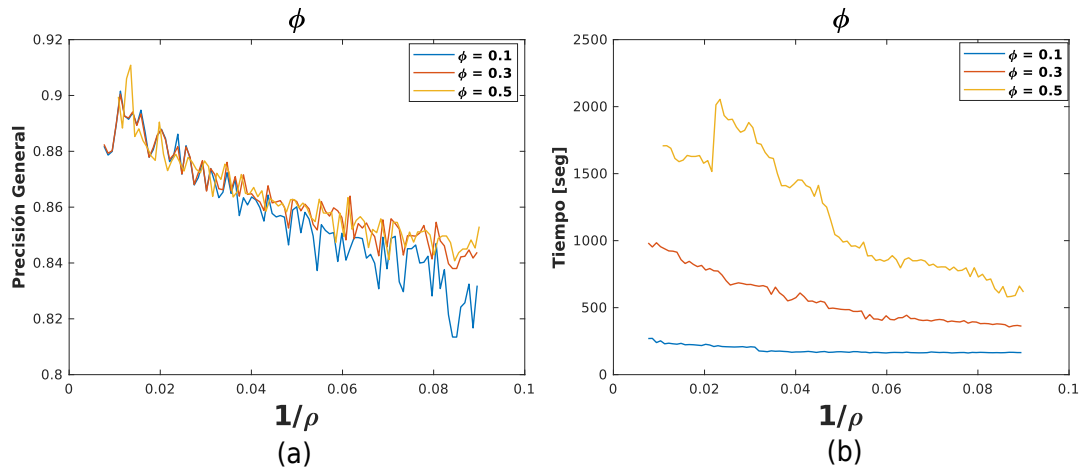


Figura 11. Influencia del parámetro  $\phi$  en la precisión general del agrupamiento en la imagen Salinas.

Como se observa en la figura 11(a) la variación del parámetro  $\phi$  no tiene un impacto significativo en la precisión del agrupamiento, pero si tiene un alto costo en el tiempo de ejecución del algoritmo, como se muestra en la figura 11(b). Teniendo en cuenta el tiempo de cómputo, y la precisión en el agrupamiento se decidió definir el parámetro  $\phi$  para todas las simulaciones como una constante con el valor de 0.1.

### 3.2. Mapas Visuales y Resultados Cuantitativos

El método de agrupación de subespacios para HSI propuesto se comparo con el algoritmo de agrupación de subespacios con restricción de escasez SSC, el algoritmo escalable de agrupación de subespacios dispersos SSSC, el algoritmo con un enfoque escalable y robusto en agrupamiento de subespacios dispersos (Abdolali et al., 2019) (SR-SSC, por sus siglas en inglés), el algoritmo escalable de agrupación escasa de subespacios escasos por búsqueda ortogonal coincidente (You

et al., 2016b) (SSC-OMP, por sus siglas en inglés), el algoritmo de conjunto activo la agrupación subespacial de red elástica escalable (You et al., 2016a) (ORGEN, por sus siglas en inglés).

En la figura 12, 13 y 14 se presentan los resultados visuales de agrupamiento sobre las imágenes utilizadas con dos de los algoritmos del estado del arte que mejor se desempeñaron en cada imagen. Los resultados cuantitativos de esta clasificación se muestran en la tabla 3. Similarmente, la figura 13 y la tabla 4 presentan los resultados visuales y cuantitativos de la universidad de Pavia. Finalmente la figura 14 y la figura 5 presentan los resultados visuales y cuantitativos de la imagen Salinas. En las tablas, el valor óptimo de cada fila se muestra en negrita y el segundo mejor resultado están subrayados. Como se muestra en las tablas 3, 4 y 5 se observó que el enfoque de agrupamiento de subespacios en imágenes espectrales propuesto proporciona resultados comparables con algunos algoritmos del estado del arte, esto debido a que nuestro método está basado en el refrán "divide y vencerás", ya que dividimos la HSI en pequeños segmentos y a estos se les aplica el enfoque del **Algoritmo 2**, sujeto a que los puntos dentro de la muestra no son elegidos aleatoriamente, son elegidos basándonos a la información espacial presente en cada segmento.

Para las imágenes de la universidad de Pavia y Salinas no se mostrarán resultados cuantitativos ni visuales de el algoritmo SSC debido a que este algoritmo expresa cada punto como una combinación lineal de todos los puntos, lo cual es ineficiente cuando el número de puntos es grande (propiedad de autoexpresividad). Este algoritmo almacena la matriz  $C$  en memoria la cual es de doble precisión y tiene dimensiones  $MN \times MN$ . Debido a esto, el algoritmo requería para la imagen Salinas aproximadamente 98 GB y aproximadamente 340 GB de memoria RAM para la universidad de Pavia. Por lo tanto, fue imposible realizar este experimento debido a la limitada

Clase	SSC	SR-SSC	EnSC	SSCOMP	SSSC	SRSSSC
Alfalfa	0	0	0	0	0	<b>97.06</b>
Maíz pequeño	57.05	53.20	59.77	16.70	<u>62.20</u>	<b>71.10</b>
Maíz mediano	18.54	14.00	<u>19.68</u>	4	19	<b>88.91</b>
Maíz grande	<u>10</u>	7	0.88	0	4.4	<b>89.17</b>
Pastizales	82.04	22.30	34.10	<b>94.10</b>	35.30	<u>83.52</u>
Xantorroeóideas	1.67	<u>82.90</u>	81.39	51.60	71.20	<b>88.58</b>
Segado de pastos	0	0	<u>4.73</u>	0.2	0	<b>44.19</b>
Heno en Hileras	5.99	85.10	85.67	27.6	<u>86.8</u>	<b>99.79</b>
Avena	0	4.7	4.7	<u>12.1</u>	3.7	<b>67.86</b>
Soja pequeña	21.34	32.40	<u>37.87</u>	37.50	29.60	<b>84.69</b>
Soja mediana	28.92	53.90	<u>56.36</u>	35.70	52.90	<b>94.81</b>
Soja limpia	<u>34.65</u>	15.30	15.68	32.10	13.20	<b>68.23</b>
Trigo	32.84	<u>62.30</u>	32.95	47.40	29.90	<b>96.65</b>
Bosque	<b>99.01</b>	72.80	87.90	50	84.10	<u>93.41</u>
Edificios-Césped-Árboles	<u>85</u>	28.40	29.43	4.300	12.60	<b>97.91</b>
Torres de piedra-acero	<b>92.65</b>	<u>73.60</u>	13.77	0	0	67.72
Precisión general (OA)	15.5	<u>36.84</u>	33.21	20.32	35.44	<b>86.15</b>
Precisión promedio (AA)	17.9	39.06	<u>40.64</u>	11.54	32.36	<b>84.83</b>
Kappa	9.46	<u>30.37</u>	27.27	9.53	29.07	<b>84.28</b>
NMI	0.197	0.423	<u>0.424</u>	0.132	0.4155	<b>0.766</b>
Tiempo [seg]	9050	<u>16.85</u>	1640	22.18	56.77	<b>15.05</b>

Tabla 3

*Resultados cuantitativos de agrupación en la imagen Indian Pines*

disponibilidad en los recursos de hardware.

En los resultados mostrados anteriormente se asume que se conoce la cantidad de grupos en cada segmento. Sin embargo, es posible que este valor no se conozca en la práctica. Como se mencionó en la sección 2.3, es posible estimar el valor de  $k_i$ . Con el objetivo de mostrar la eficiencia

Clase	SR-SSC	EnSC	SSCOMP	SSSC	SRSSSC
Asfalto	62.10	<u>70</u>	52.90	61.07	<b>94.30</b>
Prado	83.90	76	<u>87.70</u>	77.09	<b>92.95</b>
Grava	0	0	<u>27.20</u>	1.21	<b>98.32</b>
Árboles	48.40	<u>66</u>	<b>69.60</b>	58.11	61.30
Hojas de metal pintadas	<u>80.90</u>	58	0	0	<b>92.26</b>
Suelo desnudo	<u>37.70</u>	23	20.30	20.23	<b>99.52</b>
Betún	0	8	3.10	<u>13.23</u>	<b>98.95</b>
Ladrillos autobloqueantes	<u>37.80</u>	5	27.30	13.72	<b>96.01</b>
Oscuridad	<u>78.30</u>	0	0.9	3.75	<b>79.98</b>
Precisión general (OA)	<u>43.66</u>	32.98	39.66	42.7	<b>90.51</b>
Precisión promedio (AA)	<u>51.13</u>	37.18	30.11	34.3	<b>89.87</b>
Kappa	<u>34.57</u>	22.38	28.38	24.51	<b>87.39</b>
NMI	<u>0.5197</u>	0.4771	0.413	0.3685	<b>0.8027</b>
Tiempo [seg]	<b>233</b>	5626.8	1369	29571	<u>611.18</u>

Tabla 4

*Resultados cuantitativos de agrupación en la universidad de Pavia.*

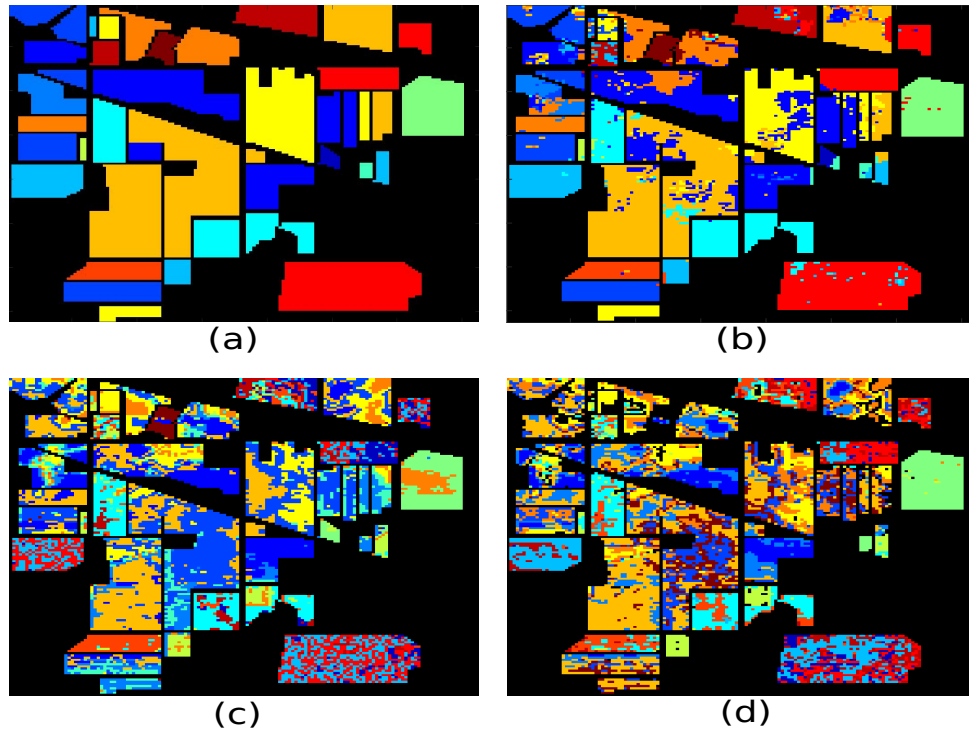


Figura 12. Resultado de agrupamiento en Indian Pines . (a) Imagen de etiquetas de las clases presentes . (b) SRSSSC con  $N_{seg} = 400$ . (c) SR-SCC. (d) SSSC.

del método propuesto bajo este escenario, a continuación se muestra el resultado de agrupamiento sobre la HSI Indian Pines cuando el valor de  $k_i$  es calculado mediante 22:

Como se puede comparar en la figura 15(a) y 16(a) el resultado de agrupamiento no varía en grandes proporciones, pero el tiempo de cómputo sí varía de forma significativa, al comparar la figura 15(b) y 16(b). Esto se debe a los cálculos adicionales que se debe aplicar sobre cada segmento para estimar el valor de  $k_i$ .

### 3.3. Análisis del rendimiento en función de $\rho$

La cantidad de superpíxeles calculados sobre la imagen  $\mathbf{B}$  tiene un impacto significativo en los resultados de agrupamiento y en el tiempo de calcular los grupos presentes en la HSI. Como

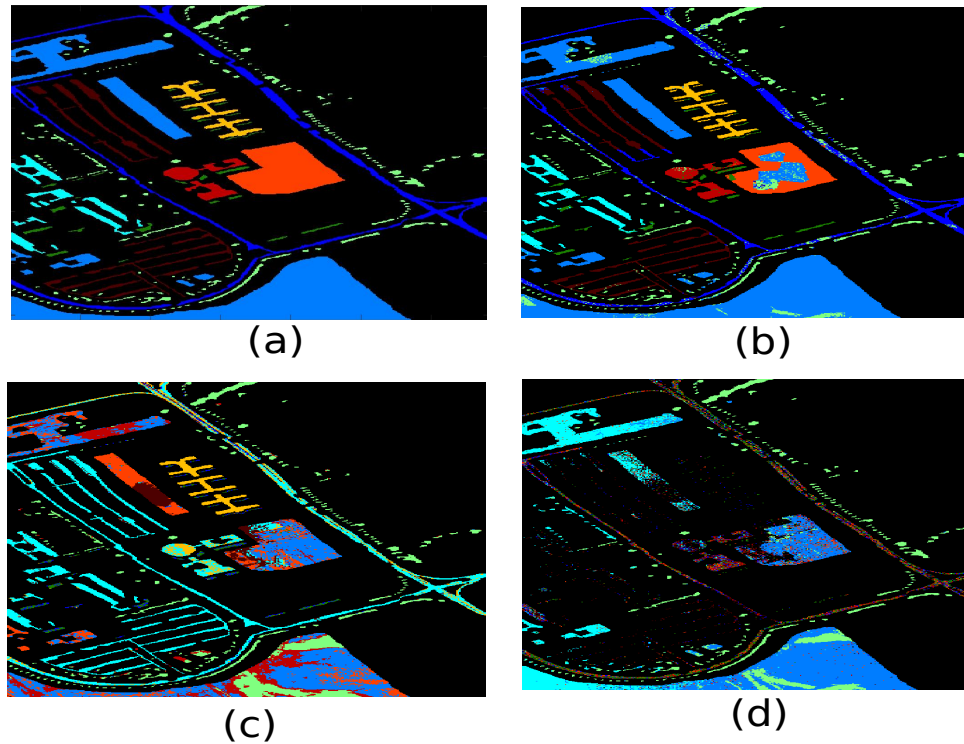


Figura 13. Resultado de agrupamiento de la universidad de Pavia. (a) Imagen de etiquetas de las clases presentes. (b) SRSSC con  $N_{seg} = 650$ . (c) SR-SCC. (d) SSSC.

se observa en las figuras 16, 17 y 18, si el valor de  $\rho$  es pequeño la precisión de cualquier método empieza a disminuir. Esto se presenta con las imágenes de alta resolución como la Salinas y la universidad de Pavia, si el valor de  $\rho$  es grande, la precisión general (OA) es baja y el tiempo de cómputo es alto. Esto se debe a que la mayoría de los puntos son agrupados por el algoritmo SSC el cual presenta una baja eficiencia como se muestra en la tabla 3 los resultados de agrupación son bajos y el tiempo de cómputo es alto.

Clase	SR-SSC	EnSC	SSCOMP	SSSC	SRSSSC
Brócoli-Hierbas Verdes 1	0	0	<u>4.50</u>	0	<b>99.95</b>
Brócoli-Hierbas Verdes 2	51.90	<u>64.30</u>	29.30	62.20	<b>99.95</b>
Barbecho	93.30	<b>100</b>	55.00	19	<u>99.19</u>
Arado áspero en barbecho	96.10	3.70	<b>99</b>	4.40	<u>98.36</u>
Barbecho liso	74.30	61.40	<u>85.60</u>	35.30	<b>96.92</b>
Rastrojo	<u>99.80</u>	<b>99.90</b>	62.20	71.20	99.27
Apio	64	<u>98</u>	18.40	0	<b>99.83</b>
Uvas	63.90	72.90	64.50	<b>86.80</b>	<u>75.29</u>
Viñedo	<u>82.70</u>	0.6000	71.80	3.700	<b>97.42</b>
Maíz verde	63.60	<u>64.20</u>	18.50	29.60	<b>99.89</b>
Lechuga Romana de 4 semanas	4.90	0	1.80	<u>52.90</u>	<b>99.90</b>
Lechuga Romana de 5 semanas	<u>91.40</u>	39.10	22.50	13.20	<b>93.58</b>
Lechuga Romana de 6 semanas	0	<u>84</u>	0	29.90	<b>90.61</b>
Lechuga Romana de 7 semanas	38.70	<b>93.40</b>	11.10	<u>84.10</u>	72.07
Viñedo sin entrenar	50	<u>51</u>	45.30	12.60	<b>99.78</b>
Viñedo enrejado verticalmente	<u>98.50</u>	96.60	1.20	0	<b>99.94</b>
Precisión general (OA)	56.98	56.18	33.36	<u>70.76</u>	<b>91.44</b>
Precisión promedio (AA)	54.79	56.23	29.45	<u>60.05</u>	<b>93.71</b>
Kappa	52.84	51.96	27.94	<u>66.04</u>	<b>90.42</b>
NMI	0.7102	0.7442	0.5274	<u>0.8097</u>	<b>0.922</b>
Tiempo [seg]	<b>139.14</b>	6660	480	2912.8	<u>263.12</u>

Tabla 5

*Resultados cuantitativos de agrupación en la imagen Salinas.*

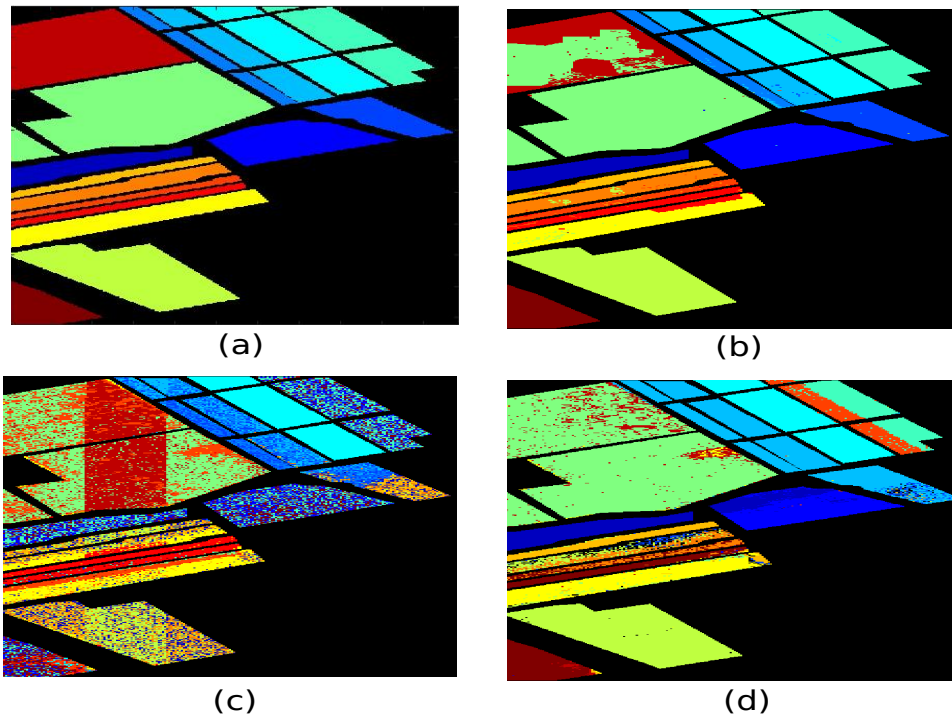


Figura 14. Resultado de agrupamiento Salinas.(a) Imagen de etiquetas de las clases presentes . (b) SRSSC con  $N_{seg} = 600$ . (c) SR-SCC. (d) SSSC.

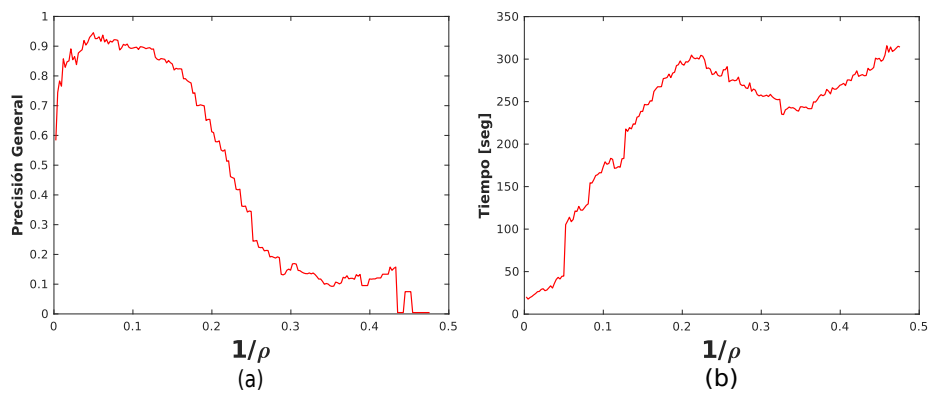


Figura 15. Resultado de agrupamiento Indian Pines calculando el valor de  $k$  sobre cada segmento. (a) Precisión general (OA). (b) Tiempo de ejecución.

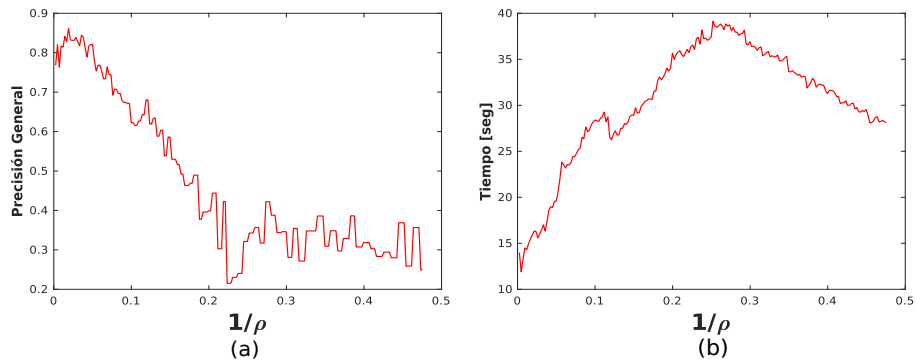


Figura 16. Resultado de agrupamiento Indian Pines.

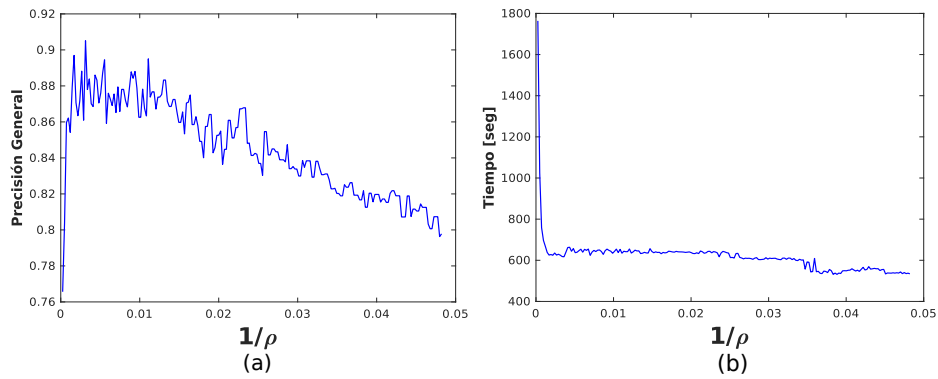


Figura 17. Resultado de agrupamiento universidad de Pavia.

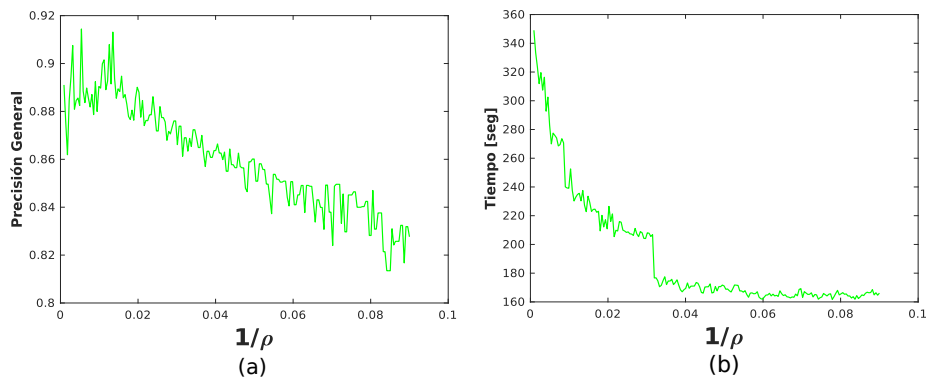


Figura 18. Resultado de agrupamiento Salinas.

#### 4. Conclusiones

En este trabajo de investigación se desarrolló un nuevo enfoque de agrupación de subespacios en imágenes espectrales, el cual tiene una baja complejidad computacional utilizando el algoritmo de agrupación de subespacios escasos (SSC). Esto se logra usando técnicas de segmentación de imágenes tales como la detección de bordes con el operador Sobel 1.3.1 y superpíxeles 1.3.2, para realizar muestreos selectivos teniendo en cuenta la similitud de los datos. Logrando así, una mayor precisión de agrupamiento del 49.31 % en la HSI Indian Pines, 46.85 % sobre la HSI Universidad de Pavia y 20.68 en la HSI salinas, esto sobre el segundo mejor resultado de los algoritmos del estado del arte con los cuales nos compararnos, utilizando el lenguaje y entorno de programación MATLAB. Otro hecho que se evidenció es que el rendimiento de nuestro método depende en la distribución de la proporción de los puntos en la muestra y fuera de la muestra.

Este trabajo también puede ampliarse o mejorarse a partir de los siguientes aspectos. Primero, se puede implementar un nuevo enfoque que aborde la agrupación de los puntos fuera de la muestra, dado que en el marco propuesto se basa en el supuesto de que los datos fuera de la muestra pueden representarse mediante datos dentro de la muestra. Por lo tanto, el método puede fallar al manejar un dato fuera de la muestra cuando proviene de un nuevo subespacio que no emerge de los datos de la muestra. En segundo lugar, se pueden explorar diferentes técnicas de extracción de los bordes en una HSI, ya que como se mencionó en la sección 1.3 detectar bordes en una HSI es una tarea que aún no está bien establecida.

### Referencias Bibliográficas

- Abdolali, M., Gillis, N., and Rahmati, M. (2019). Scalable and robust sparse subspace clustering using randomized clustering and multilayer graphs. *Signal Processing*, 163:166–180.
- Achanta, R., Shaji, A., Smith, K., Lucchi, A., Fua, P., and Süsstrunk, S. (2012). Slic superpixels compared to state-of-the-art superpixel methods. *IEEE transactions on pattern analysis and machine intelligence*, 34(11):2274–2282.
- Al-Amri, S. S., Kalyankar, N. V., et al. (2010). Image segmentation by using threshold techniques. *arXiv preprint arXiv:1005.4020*.
- Boser, B. E., Guyon, I. M., and Vapnik, V. N. (1992). A training algorithm for optimal margin classifiers. In *Proceedings of the fifth annual workshop on Computational learning theory*, pages 144–152. ACM.
- Chen, G. and Lerman, G. (2009). Spectral curvature clustering (scc). *International Journal of Computer Vision*, 81(3):317–330.
- Duda, R. O., Hart, P. E., and Stork, D. G. (2012). *Pattern classification*. John Wiley & Sons.
- Eismann, M. T., Stocker, A. D., and Nasrabadi, N. M. (2009). Automated hyperspectral cueing for civilian search and rescue. *Proceedings of the IEEE*, 97(6):1031–1055.
- Elhamifar, E. and Vidal, R. (2009). Sparse subspace clustering. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 2790–2797. IEEE.

- Elhamifar, E. and Vidal, R. (2013). Sparse subspace clustering: Algorithm, theory, and applications. *IEEE transactions on pattern analysis and machine intelligence*, 35(11):2765–2781.
- Gao, L., Kester, R. T., Hagen, N., and Tkaczyk, T. S. (2010). Snapshot image mapping spectrometer (ims) with high sampling density for hyperspectral microscopy. *Optics express*, 18(14):14330–14344.
- Goh, A. and Vidal, R. (2007). Segmenting motions of different types by unsupervised manifold clustering. In *2007 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–6. IEEE.
- González, C. (1996). R.; e. woods, r., tratamiento digital de imágenes.
- Halko, N., Martinsson, P.-G., and Tropp, J. A. (2011). Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions. *SIAM review*, 53(2):217–288.
- Haralick, R. M. and Shapiro, L. G. (1992). *Computer and robot vision*, volume 1. Addison-wesley Reading.
- He, W., Zhang, H., Zhang, L., and Shen, H. (2015). Hyperspectral image denoising via noise-adjusted iterative low-rank matrix approximation. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 8(6):3050–3061.
- Hinojosa Montero, C. A. (2018). *Coded Aperture Design For Compressive Spectral Imaging Subspace Clustering*. PhD thesis, Universidad Industrial de Santander, Escuela De Ing. De Sistemas.

- Hörig, B., Kühn, F., Oschütz, F., and Lehmann, F. (2001). Hymap hyperspectral remote sensing to detect hydrocarbons. *International Journal of Remote Sensing*, 22(8):1413–1422.
- Kerekes, J. P. and Schott, J. R. (2007). Hyperspectral imaging systems. *Hyperspectral data exploitation: Theory and applications*, pages 19–45.
- Li, Z., Wu, X.-M., and Chang, S.-F. (2012). Segmentation using superpixels: A bipartite graph partitioning approach. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 789–796. IEEE.
- Ma, Y., Derksen, H., Hong, W., and Wright, J. (2007). Segmentation of multivariate mixed data via lossy data coding and compression. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, (9):1546–1562.
- Macedo-Cruz, A., Pajares-Martinsanz, G., and Santos-Peñas, M. (2010). Clasificación no supervisada con imágenes a color de cobertura terrestre. *Agrociencia*, 44(6):711–722.
- Manolakis, D. and Shaw, G. (2002). Detection algorithms for hyperspectral imaging applications. *IEEE Signal Processing Magazine*, 19(1):29–43.
- MUÑOZ, A., OSORIO, G., PRIETO, F., and ANGULO, F. (2001). Detección de patologías celulares empleando visión artificial. *Grupo de Percepción y Control Inteligente, Universidad Nacional de Colombia, Manizales*.
- Ng, A. Y., Jordan, M. I., Weiss, Y., et al. (2002). On spectral clustering: Analysis and an algorithm. *Advances in neural information processing systems*, 2:849–856.

- Osorio, J. A. C., Muriel, A., and Vargas, J. A. M. (2011). Comparación cualitativa y cuantitativa de las técnicas básicas de umbralización global basadas en histogramas para el procesamiento digital de imágenes. *Scientia et Technica*, 3(49):266–272.
- Palomino, N. L. S. and Concha, U. N. R. (2009). Técnicas de segmentación en procesamiento digital de imágenes. *Revista de investigación de Sistemas e Informática*, 6(2):9–16.
- Patel, N., Patnaik, C., Dutta, S., Shekh, A., and Dave, A. (2001). Study of crop growth parameters using airborne imaging spectrometer data. *International Journal of Remote Sensing*, 22(12):2401–2411.
- Peng, B., Zhang, L., and Zhang, D. (2011). Automatic image segmentation by dynamic region merging. *IEEE Transactions on Image Processing*, 20(12):3592–3605.
- Peng, X., Tang, H., Zhang, L., Yi, Z., and Xiao, S. (2013). A unified framework for representation-based subspace clustering of out-of-sample and large-scale data.
- Ren, X. and Malik, J. (2003). Learning a classification model for segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition*, page 10. IEEE.
- Shapiro, L. and George, C. (2002). Stockman g: computer vision. In *Prentice Hall*.
- Shaw, G. A. and Burke, H.-H. K. (2003). Spectral imaging for remote sensing. *Lincoln Laboratory Journal*, 14(1):3–28.

- Steinbach, M., Ertöz, L., and Kumar, V. (2004). The challenges of clustering high dimensional data. In *New directions in statistical physics*, pages 273–309. Springer.
- Tipping, M. E. and Bishop, C. M. (1999). Mixtures of probabilistic principal component analyzers. *Neural computation*, 11(2):443–482.
- Trucco, E. and Verri, A. (1998). *Introductory techniques for 3-D computer vision*, volume 201. Prentice Hall Englewood Cliffs.
- Umbaugh, S. E. (2010). *Digital image processing and analysis: human and computer vision applications with CVIPtools*. CRC press.
- Vapnik, V. (2013). *The nature of statistical learning theory*. Springer Science & Business Media.
- Velasco, A. C., García, C. A. V., Morales, F. A. R., Castelblanco, S. F. C., and Fuentes, H. A. (2015). Aplicaciones y retos del sensado remoto hiperespectral en la geología colombiana. *Facultad de Ingeniería*, 24(40):17–29.
- Vidal, R. (2011). Subspace clustering. *IEEE Signal Processing Magazine*, pages 52–68.
- Vidal, R., Ma, Y., and Sastry, S. (2005). Generalized principal component analysis (gpca). *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(12):1945–1959.
- Von Luxburg, U. (2007). A tutorial on spectral clustering. *Statistics and computing*, 17(4):395–416.

- Yan, J. and Pollefeys, M. (2006). A general framework for motion segmentation: Independent, articulated, rigid, non-rigid, degenerate and non-degenerate. In *European conference on computer vision*, pages 94–106. Springer.
- You, C., Li, C.-G., Robinson, D. P., and Vidal, R. (2016a). Oracle based active set algorithm for scalable elastic net subspace clustering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3928–3937.
- You, C., Robinson, D., and Vidal, R. (2016b). Scalable sparse subspace clustering by orthogonal matching pursuit. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3918–3927.
- Zelnik-Manor, L. and Irani, M. (2003). Degeneracies, dependencies and their implications in multi-body and multi-sequence factorizations. In *Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on*, volume 2, pages II–287. IEEE.
- Zhang, H., Zhai, H., Zhang, L., and Li, P. (2016). Spectral–spatial sparse subspace clustering for hyperspectral remote sensing images. *IEEE Transactions on Geoscience and Remote Sensing*, 54(6):3672–3684.
- Zhang, T., Szlam, A., and Lerman, G. (2009). Median k-flats for hybrid linear modeling with many outliers. In *Computer Vision Workshops (ICCV Workshops), 2009 IEEE 12th International Conference on*, pages 234–241. IEEE.

Zhang, T., Szlám, A., Wang, Y., and Lerman, G. (2012). Hybrid linear modeling via local best-fit flats. *International journal of computer vision*, 100(3):217–240.

## Apéndices

### Apéndice A. ADMM para resolver el problema de optimización del agrupamiento de subespacio propuesto

En esta sección, se muestra la resolución del problema de optimización escasa mostrado en la ecuación 21. Introduciendo el método de dirección alterna de multiplicadores (ADMM, por sus siglas en inglés). Podemos aplicar la restricción de igualdad en 21 y eliminar  $Z$  del problema de optimización y resolver de manera equivalente:

$$\begin{aligned} \min_{C,E} & \|C\|_1 + \lambda_e \|E\|_1 + \frac{\lambda_z}{2} \|D - DC - E\|_F^2 \\ \text{s.t.} & \quad C^\top \mathbf{1} = \mathbf{1}, \quad \text{diag}(C) = \mathbf{0}. \end{aligned} \tag{24}$$

El procedimiento general del algoritmo ADMM es introducir variables auxiliares apropiadas en el programa de optimización, aumentar las restricciones en la función objetivo e iterativamente minimizar el lagrangiano con respecto a las variables primarias y maximizarlo con respecto a los multiplicadores de Lagrange. Con un abuso de notación, a lo largo de esta sección, denotamos por  $\text{diag}(C)$  tanto un vector cuyos elementos son las entradas diagonales de  $C$  como una matriz diagonal cuyos elementos diagonales son las entradas diagonales de  $C$ .

Para comenzar, se utiliza una matriz auxiliar  $A \in \mathbb{R}^{N \times N}$  para que el problema quede así:

$$\begin{aligned} \min_{C,E,A} & \|C\|_1 + \lambda_e \|E\|_1 + \frac{\lambda_z}{2} \|D - DA - E\|_F^2 \\ \text{s.t.} & \quad A^\top \mathbf{1} = \mathbf{1}, \quad A = C - \text{diag}(C), \end{aligned} \quad (25)$$

cuya solución para  $(C, E)$  coincide con la solución de 24, la introducción de  $A$  ayuda a obtener actualizaciones eficientes en las variables de optimización. Luego, usando un parámetro  $\rho > 0$ , agregamos a la función 25 dos términos de penalidad correspondientes a las restricciones  $A^\top \mathbf{1} = \mathbf{1}$  y  $A = C - \text{diag}(C)$ , queda por resolver el siguiente problema de optimización:

$$\begin{aligned} \min_{C,E,A} & \|C\|_1 + \lambda_e \|E\|_1 + \frac{\lambda_z}{2} \|D - DA - E\|_F^2 \\ & + \frac{\rho}{2} \|A^\top \mathbf{1} - \mathbf{1}\|_2^2 + \frac{\rho}{2} \|A - (C - \text{diag}(C))\|_F^2 \\ \text{s.t.} & \quad A^\top \mathbf{1} = \mathbf{1}, \quad A = C - \text{diag}(C). \end{aligned} \quad (26)$$

Al agregar los términos de penalización a 25 no cambia su solución óptima, es decir, tanto 25 como 26 tienen las mismas soluciones, ya que para cualquier solución factible de 26 que satisfaga las restricciones, los términos de penalización desaparecen. Sin embargo, agregar los términos de penalización hace que la función de optimización sea estrictamente convexa en términos de las variables de optimización  $(C, E, A)$ , que permite usar el enfoque **ADMM**.

Introduciendo un vector  $\delta \in \mathbb{R}^N$  y una matriz  $\Delta \in \mathbb{R}^{N \times N}$  de multiplicadores de Lagrange para las dos restricciones de igualdad en 26, se puede escribir la ecuación 26 en términos lagran-

gianos:

$$\begin{aligned}
 \mathcal{L}(C, A, E, \delta, \Delta) = & \|C\|_1 + \lambda_e \|E\|_1 + \frac{\lambda_z}{2} \|D - DA - E\|_F^2 \\
 & + \frac{\rho}{2} \|A^\top \mathbf{1} - \mathbf{1}\|_2^2 + \frac{\rho}{2} \|A - (C - \text{diag}(C))\|_F^2 \\
 & + \delta^\top (A^\top \mathbf{1} - \mathbf{1}) + \text{tr}(\Delta^\top (A - C + \text{diag}(C))),
 \end{aligned} \tag{27}$$

donde  $\text{tr}(\cdot)$  denota el operador de traza de una matriz dada. El enfoque ADMM consiste en un procedimiento iterativo como el siguiente: Denotamos por  $(C^{(k)}, E^{(k)}, A^{(k)})$  las variables de optimización en la iteración  $k$  y  $(\delta^{(k)}, \Delta^{(k)})$  los multiplicadores de Lagrange en la iteración  $k$ :

- Obteniendo  $A^{(k+1)}$  minimizando  $\mathcal{L}$  con respecto a  $A$ , mientras  $(C^{(k)}, E^{(k)}, \delta^{(k)}, \Delta^{(k)})$  son fijos. Tenga en cuenta que calcular la derivada de  $\mathcal{L}$  con respecto a  $A$  e igualándolo a cero, obtenemos:

$$\begin{aligned}
 (\lambda_z D^\top D + \rho I + \rho \mathbf{1}\mathbf{1}^\top) A^{(k+1)} = & \lambda_z D^\top (D - E^{(k)}) \\
 & + \rho (\mathbf{1}\mathbf{1}^\top + C^{(k)}) - \mathbf{1} \delta^{(k)\top} - \Delta^{(k)}.
 \end{aligned} \tag{28}$$

En otras palabras,  $A^{(k+1)}$  se obtiene resolviendo un sistema lineal de  $N \times N$  ecuaciones. Cuando  $N$  no es muy grande, uno puede simplemente invertir la matriz para obtener  $A^{(k+1)}$  desde 28. Para grandes valores de  $N$  se deben emplear métodos de gradiente conjugado para resolver  $A^{(k+1)}$ .

- Obteniendo  $C^{(k+1)}$  minimizando  $\mathcal{L}$  respecto a  $C$ , mientras  $(A^{(k)}, E^{(k)}, \delta^{(k)}, \Delta^{(k)})$  son fijos.

Se ve que la actualización de  $C$  también tiene una solución de forma cerrada dada por:

$$\begin{aligned}
 C^{(k+1)} &= J - \text{diag}(J), \\
 J &\triangleq \mathcal{T}_{\frac{1}{\rho}} \left( A^{(k+1)} + \Delta^{(k)} / \rho \right),
 \end{aligned} \tag{29}$$

donde  $\mathcal{T}_{\eta}(\cdot)$  es el operador de umbral de contracción que actúa sobre cada elemento de la matriz dada y se define como:

$$\mathcal{T}_{\eta}(v) = (|v| - \eta)_+ \text{sgn}(v).$$

El operador  $(\cdot)_+$  devuelve su argumento si no es negativo y devuelve cero de lo contrario.

- Obteniendo  $E^{(k+1)}$  minimizando  $\mathcal{L}$  respecto a  $E$ , mientras  $(C^{(k+1)}, A^{(k+1)}, \delta^{(k)}, \Delta^{(k)})$  son fijos. La actualización de  $E$  también se puede calcular en forma cerrada como:

$$E^{(k+1)} = \mathcal{T}_{\frac{\lambda_e}{\lambda_c}} \left( DA^{(k+1)} - D \right). \tag{30}$$

- Teniendo  $C^{(k+1)}, A^{(k+1)}, E^{(k+1)}$  fijos, aplicar el gradiente ascendente con el tamaño de paso de  $\rho$  en los multiplicadores de Lagrange:

$$\begin{aligned}
 \delta^{(k+1)} &= \delta^{(k)} + \rho \left( A^{(k+1)\top} \mathbf{1} - \mathbf{1} \right), \\
 \Delta^{(k+1)} &= \Delta^{(k)} + \rho \left( A^{(k+1)} - C^{(k+1)} \right).
 \end{aligned} \tag{31}$$

Estos tres pasos se repiten hasta que se logre la convergencia o el número de iteraciones exceda un número máximo de iteraciones. La convergencia se logra cuando tenemos

$\left\|A^{(k)\top} \mathbf{1} - \mathbf{1}\right\|_{\infty} \leq \varepsilon$ 
 $\left\|A_t^{(k)} - C^{(k)}\right\|_{\infty} \leq \varepsilon$ ,
 $\left\|A^{(k)} - A^{(k-1)}\right\|_{\infty} \leq \varepsilon$ 
y
 $\|E^{(\vec{k})} - E^{(k-1)}\|_{\infty} \leq \varepsilon$ ,
donde  $\varepsilon$  denota la tolerancia al error para los residuos primarios y duales. En la práctica, la elección de  $\varepsilon = 10^{-4}$  funciona bien en experimentos reales.

## Apéndice B. Métricas de Evaluación

Para comparar el rendimiento de agrupación de nuestro método, utilizamos cinco métricas estándar: precisión del usuario, precisión promedio, precisión general, información mutua normalizada (UA,AA,OA,NMI) respectivamente por sus siglas en inglés y coeficiente Kappa. En particular, los coeficientes UA, AA, OA y Kappa se pueden obtener por medio de una matriz de error (también conocida como matriz de confusión). Las matrices de error comparan, categoría por categoría, la relación entre los datos de referencia conocidos (matriz de etiquetas) y los resultados correspondientes de una clasificación automatizada. Dichas matrices son cuadradas, con el número de filas y columnas igual al número de categorías cuya precisión de clasificación se está evaluando.

- **Precisión general:** se calcula dividiendo el número total de píxeles clasificados correctamente (es decir, la suma de los elementos a lo largo de la diagonal principal) por el número total de píxeles de referencia.
- **Precisión de usuario (precisión individual de las clases):** se calcula dividiendo el número total de píxeles clasificados correctamente (es decir, la suma de los elementos a lo largo de la diagonal principal) por el número total de píxeles de referencia.
- **Precisión promedio:** Es el promedio de la precisión de usuario.
- **Coficiente Kappa:** es una medida de la diferencia entre el acuerdo real entre los datos de referencia y un clasificador automatizado y el acuerdo casual entre los datos de referencia y un clasificador aleatorio. Específicamente dada una matriz de error  $\mathbf{Q}$  con  $r_q$  filas y  $N_q$

número de observaciones, el coeficiente kappa viene dado por:

$$\text{Kappa} = \frac{N_q \sum_{i=1}^{r_q} Q_{ii} - \sum_{i=1}^{r_q} (Q_{i+} \cdot Q_{+i})}{N_q^2 - \sum_{i=1}^{r_q} (Q_{i+} \cdot Q_{+i})}, \quad (32)$$

donde  $Q_{ii}$  es el número de observaciones en una fila  $i$  y columna  $i$ ,  $Q_{i+}$  es el número total de observaciones en la fila  $i$ ,  $Q_{+i}$  es el total de observaciones en la columna  $i$ .

- Información mutua normalizada:** Es una normalización de la información mutua, el cual consiste en valores entre 0 (no hay información mutua) y 1 (hay correlación perfecta). La información mutua mide la cantidad de información por la cual nuestro conocimiento sobre las clases aumenta cuando se nos dice cuáles son los grupos. Específicamente, dado las etiquetas de las clases  $\Gamma$  y las etiquetas de agrupamiento  $\Omega$  tenemos:

$$\text{NMI}(\Omega, \Gamma) = \frac{\text{MI}(\Omega; \Gamma)}{[H(\Omega) + H(\Gamma)]/2}, \quad (33)$$

con

$$\text{MI} = \sum_k \sum_j P(\omega_k \cap \gamma_j) \log \frac{P(\omega_k \cap \gamma_j)}{P(\omega_k) P(\gamma_j)}, \quad (34)$$

donde  $P(\omega_k)$ ,  $P(\gamma_j)$ , y  $P(\omega_k \cap \gamma_j)$  son las probabilidades de que un punto de datos esté en un grupo  $\omega_j$ , clase  $\gamma_j$ , y en la intersección de  $\omega_k$ , clase  $\gamma_k$  respectivamente. La función  $H(\cdot)$

en 33 es la entropía definida por:

$$H(\Omega) = - \sum_k P(\omega_k) \log P(\omega_k). \quad (35)$$