Zhang *et al.*
EURASIP Journal on Advances in Signal Processing (2022) 2022:92
https://doi.org/10.1186/s13634-022-00926-8

EURASIP Journal on Advances in Signal Processing

RESEARCH Open Access

Multilayer graph spectral analysis for hyperspectral images

Songyang Zhang, Qinwen Deng and Zhi Ding*

*Correspondence:
zding@ucdavis.edu

Department of Electrical and Computer Engineering, University of California at Davis, Davis, USA

Abstract

Hyperspectral imaging has broad applications and impacts in areas including environmental science, weather, and geo/space exploration. The intrinsic spectral–spatial structures and potential multi-level features in different frequency bands make multilayer graph an intuitive model for hyperspectral images (HSI). To study the underlying characteristics of HSI and to take the advantage of graph signal processing (GSP) tools, this work proposes a multilayer graph spectral analysis for hyperspectral images based on multilayer graph signal processing (M-GSP). More specifically, we present multilayer graph (MLG) models and tensor representations for HSI. By exploring multilayer graph spectral space, we develop MLG-based methods for HSI applications, including unsupervised segmentation and supervised classification. Our experimental results demonstrate the strength of M-GSP in HSI processing and spectral–spatial information extraction.

Keywords: Multilayer graph signal processing, Hyperspectral imaging, Segmentation

1 Introduction

1.1 Background and significance

Hyperspectral imaging is an analytical technique operating on images at different wavelengths for given geographical areas [1–3]. Exploiting a wealth of spectral–spatial information, hyperspectral images (HSIs) have seen broad applications in areas such as urban mapping, environment management, crop analysis and food safety inspection [4–6]. Among a variety of hyperspectral imaging analysis for these applications, graph-based approaches have recently attracted substantial attentions due to its power in uncovering the underlying spatial structures of HSI. Modeling pixels as nodes and their internal interactions as edges, graph models can be constructed to capture the geometric information of hyperspectral images. Advanced tools, such as graph signal processing (GSP) [7] and graph convolutional networks (GCN) [8], can next be applied in HSI processing tasks, including coding [9], classification [10] and reconstruction [11].

Despite the successes in hyperspectral imaging analysis, most of the graph-based methods focus primarily on spatial geometry and consider the same graph connections for the multiple spectral bands, shown as Fig. 1b. However, such a single-layer graph model fails to explore different spectrum features of each individual band. For

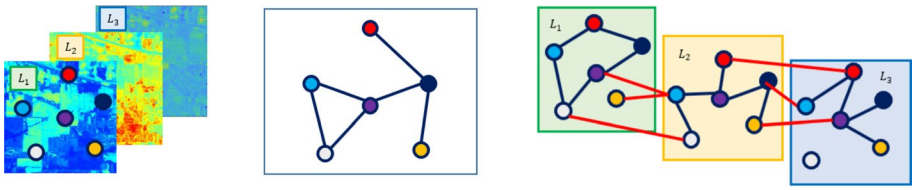


Fig. 1 Examples of graph models for HSI: **a** Stack of 3-band images in Indian Pines where six pixels/superpixels are selected as examples of nodes; **b** A single-layer graph model for the 3-band HSI in (**a**) based on feature similarity; **c** Example of a multilayer graph model with 3 layers

example, Fig. 1a shows that different spectral bands may display different distributions of the pixel volumes, which could mean different spatial structures. The high dimensionality of HSIs and their heterogeneous underlying structures for different bands do not lend themselves to traditional single-layer graphs. A more generalized geometric model needs to be explored to represent HSI efficiently.

To capture the complex spectral–spatial interactions among pixels, recent attention has been attracted in a high-dimensional graph named as multilayer networks (also known as multilayer graphs) [12]. In a multilayer network, nodes are grouped into meaningful clusters called "layers" depending on their features. Different layers could have heterogeneous graph structures and interact with each other. An example of multilayer network (graph) is shown in Fig. 1c. Modeling each spectral frame as layers, the multilayer graph (MLG) can be an intuitive model to capture the spectral–spatial interactions. Now, one major problem is how to generalize traditional graph-based analysis to multilayer graphs and process the high-dimensional multilayer structures efficiently. To compensate, some works would represent each layer with an individual graph while neglecting the interlayer correlations [13]. Although one can also construct interlayer connections for spectral frames [9], the spectral (interlayer) and spatial (intralayer) connections are still processed separately [14], where certain joint spectral–spatial information in HSI can be neglected. Some works also consider a two-step analysis for MLG. For example, in [15], a two-step transform is proposed, where graph Fourier transform (GFT) is applied first to interlayer connections and then to interlayer interactions. However, the interlayer and intralayer connections are still processed separately in different graph Fourier spaces. For these reasons, one untackled challenge is how to capture such heterogeneous spectral–spatial structures for the MLG model of HSI in an integrative manner jointly. Recently, a tensor-based framework of multilayer graph signal processing (M-GSP) is proposed in [16] to analyze the inter- and intra-layer connections. Defining MLG spectral space based on tensor decomposition, MLG spectral analysis can be implemented for HSI, which allows both joint and individual analysis for spectral–spatial interactions in HSI by incorporating the information of inter- and intra-layer connections.

Our goal is to explore MLG in HSI processing and provide a guideline for analyzing the high-dimensional spectral–spatial correlations of HSI based on M-GSP. In this work, we revisit the M-GSP framework and introduce MLG-based spectral analysis for hyperspectral imaging. More specifically, we first introduce the models and

algebraic representation of HSI in M-GSP. Next, we develop M-GSP spectral analysis to extract features for HSI analysis. To illustrate the power of MLG in HSI, we investigate the applications of M-GSP in the feature extraction for both unsupervised HSI segmentation and supervised HSI classification as guidelines of MLG-based HSI analysis. We summarize our contributions in this work as follows:

- We introduce an MLG model, together with an alternative singular space for HSI analysis.
- We provide guidelines of applying M-GSP in HSI and suggest several application examples with novel M-GSP algorithms:
  - We develop an unsupervised HSI segmentation method based on the M-GSP spectral clustering as an application example.
  - We derive an MLG-based method for supervised HSI classification by jointly utilizing multi-resolution information. We further propose several novel schemes for decision fusion of the results from different resolutions.

We test our algorithms using the widely used Indian Pines dataset, Pavia University dataset and Salinas dataset. Our experimental results demonstrate the strength of M-GSP in modeling some spectral–spatial structures in HSI, and the efficiency of the proposed HSI segmentation algorithms.

We organize the rest of this manuscript as follows. Starting with the introduction of MLG model and M-GSP representations of HSI in Sect. 2.1, we investigate unsupervised HSI segmentation based on M-GSP spectral clustering in Sect. 2.2. Next, we introduce the M-GSP framework of supervised HSI classification in Sect. 2.3, and further propose several methods for the decision fusion of different resolutions. Section 3 presents details on the experiments and results of the proposed methods. Finally, we summarize our works in Sect. 4.

1.2 Related work

In this section, we provide an overview on graph signal processing and graph learning in hyperspectral images.

1.2.1 Graph signal processing and graph convolutional networks

Graph signal processing (GSP) has emerged as an exciting and promising new paradigm for processing large datasets with complex structures, owing to its power to extract underlying relationships among signals [7, 17]. Modeling data points and their interactions as a graph, a graph Fourier space can be defined according to the eigenspace of a graph representation for data analysis [18] including image processing [19], point cloud resampling [20] and video compression [21]. The framework of GSP is further generalized with fundamentals within the graph Fourier space, such as graph Fourier transform [22], sampling theory [23], graph wavelet [24] and stationary processing [25]. Beyond normal graphs, GSP has also been considered for high-dimensional graphs, such as multilayer graphs [16], hypergraphs [26] and simplicial complexes [27]. In addition, Graph

neural networks (GNN) and graph convolutional networks (GCN) have become important tools in data analysis [8].

1.2.2 Graph-based hyperspectral image analysis

With the development of signal processing and learning over graphs, graph-based tools have attracted significant attentions in HSI analysis. In [9], a coding method is developed for HSI based on graph wavelets. Inspired by graph spectral transform, GSP-based methods have also been developed for the compression of hyperspectral scenes [28]. Exploring spectral–spatial features of HSI in graph spectral space, a feature extraction algorithm is proposed using collaboration-competition graphs [29]. In addition to basic GSP analysis, graph learning also has shown promise in hyperspectral imaging. For example, leveraging graph spectral convolution, GCN becomes an important tool in HSI applications, including HSI classification and reconstruction [10, 11, 30–32]. In addition to GCNs, a semi-supervised hyperspectral dimensionality reduction approach is introduced in [33] by propagating labels on a learnable graph. In [34], a novel semi-supervised cross-modality learning framework is introduced for HSI analysis with a learnable manifold. Other graph-based learning frameworks for HSI include X-ModalNet [35] and Deep-hybrid [36]. Interested readers are referred to a recent survey [37] to explore further.

2 Methods

In this section, we introduce the models and representations for hyperspectral images within the M-GSP framework. Then, we will introduce the M-GSP-based approaches for HSI unsupervised segmentation and supervised classification.

To avoid confusion of the term "network" because of its different meanings in communication and deep learning, we will use the less ambiguous term of "multilayer graph" instead of "multilayer network" in the remaining of this work.

2.1 Models and representations for hyperspectral images

We first introduce the MLG models and representations for HSI.

2.1.1 Superpixel segmentation for HSI

Before venturing into the M-GSP analysis, we first introduce the superpixel segmentation for HSI. In traditional graph-based HSI analysis, image pixels act as nodes and their pair-wise distances are calculated to form a graph [38]. However, given a large number of pixels, it becomes inefficient and sometimes impossible to implement full graph-based analysis for pixel-based HSIs. Practically, since pixels within a small region may share similar features, grouping neighboring pixels into superpixels could be a more practical way for graph construction.

In general, a suitable superpixel segmentation algorithm for HSI classification should exhibit low computation complexity and accurate detection of the object boundaries [39]. One category of superpixel segmentation in HSI uses image features such as brightness, color and texture cues, to estimate the location of segment boundaries. In [40], a superpixel estimation is developed by adopting the ultrametric contour map (UCM) approaches to the hyperspectral volumes. In [41], the band smoothness is jointly

considered with general features to group pixels. Graph-based segmentation approaches are also common in superpixel segmentation [42]. In [43], an eigen-based solution to normalized cuts (NCuts) is used for superpixel group. However, such eigen-based methods tend to suffer from time-consuming graph construction and matrix decomposition.

For more efficient superpixel segmentations, our work in this manuscript considers the entropy rate superpixel segmentation (ERS). Compared to other superpixel construction algorithms, ERS is very efficient as it only takes about 2.5 seconds to segment an image of size 481×321, while achieving superior performances in terms of standard metrics such as undersegmentation errors [44]. In ERS [44], a dataset is modeled as a graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$, in which the pixels serve as the nodes $\mathcal{V}$ and their pairwise similarities are represented by edges $\mathcal{E}$. Next, a subgraph $\mathcal{A} = \{\mathcal{V}, \mathcal{L}\}$ is formed by choosing a subset of edges $\mathcal{L} \subseteq \mathcal{E}$, such that $\mathcal{A}$ consists of fewer connected components. To obtain such a subgraph, the problem can be formulated as

$$\mathcal{L}^* = \arg_{\mathcal{L}} \max \ \mathrm{Tr}\{H(\mathcal{L}) + \alpha T(\mathcal{L})\} \tag{1}$$

$$s.t. \quad \mathcal{L} \subseteq \mathcal{E}, \tag{2}$$

in which entropy rate term $H(\mathcal{L})$ favors more compact clusters, and regularizing term $T(\mathcal{L})$ punishes large cluster size. The balancing parameter is given by $\alpha = \beta K \epsilon$ where $\epsilon = 0.5$, K is the number of superpixels, and β is the calculated based on the maximal entropy rate increase. Interested readers may refer to [39,44] for more details about the design of $H(\mathcal{L})$ and $T(\mathcal{L})$, as well as other details on ERS algorithm. Based on the objective function of Eq. (1), a greedy algorithm can be implemented to solve the problem.

ERS can segment pixels into superpixels for M-GSP with low complexity and good efficiency. From the constructed superpixels, we average the positions and features of all the pixels within each superpixel as its new position and feature for MLG construction. Note that, our goal is to provide a guideline of applying M-GSP in HSI and show M-GSP can achieve robust performance even with simple methods. We adopt ERS superpixel construction and averaging-based feature aggregation in this work. We plan to consider other advanced approaches for superpixel construction and feature generation in future works.

2.1.2 Multilayer graph construction for HSI datasets

With superpixel-represented HSI, we now begin multilayer graph construction outlined as Fig. 2. An HSI $\mathbf{X} \in \mathbb{R}^{K \times N}$, containing K spectral frames and N superpixels, can be modeled by a multilayer graph with M layers and N nodes in each layer. Specifically, the MLG consists of the following attributes

- *Layers*: To construct a MLG, we define layers based on the spectral bands. Since different spectral frames may share similar features, we first divide the bands into M clusters, i.e., $\mathbf{X}_i \in \mathbb{R}^{K_i \times N}$, $i = 1, \cdots, M$ and $\sum_{i=1}^{M} K_i = K$. Next, each cluster serves as one layer in the multilayer graph. Various clustering methods can generate features $\mathbf{X}_i$ for layer i. For example, one can divide spectral band based on a range of wavelengths. To capture correlation across different bands more efficiently, the k-means clustering is applied for band division.

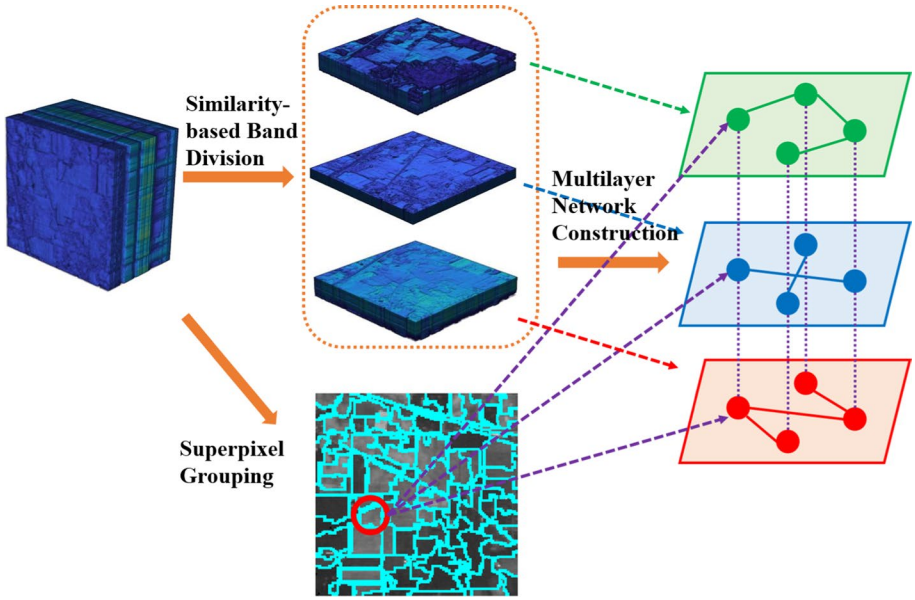


Fig. 2 Scheme of MLG-based unsupervised HSI segmentation

- *Nodes*: Shown as Fig. 2, the multilayer graph can be viewed as embedding N superpixels $\{x_1, x_2, \cdots, x_N\}$ into M layers $\{l_1, \cdots, l_M\}$. Then, we can form a multilayer graph with M layers and N nodes in each layer. We characterize each superpixel using the divided attributes, i.e., $\mathbf{X}_{\alpha,j} \in \mathbb{R}^{K_i}$ for the superpixel j's embedded node in layer α.
- *Interlayer connections*: For interlayer connections, each embedded node is connected to its counterparts in other layers, i.e., fully connected for all the embedded nodes of the same superpixel (multiplex structure [12]). Denote the weight of edge between superpixel j's embedded node in layer β and superpixel i's embedded node in layer α as $A_{\alpha i \beta j}$. The weights of interlayer connections can be calculated as

$$A_{\alpha i \beta j} = \begin{cases} 1, & \alpha \neq \beta, i = j; \\ 0, & \text{otherwise}, \end{cases} \tag{3}$$

  where each term indicates link presence. One can also assign weights to $A_{\alpha i \beta i}$ based on the feature similarities. Note that, here we introduce the weighted multiplex structure as an example for interlayer connections. Our M-GSP framework is applicable to various interlayer structures. Readers could construct graphs depending on the specific dataset and learning task.
- *Intralayer connections*: For the intralayer connections, we calculate the weights between the embedded nodes of superpixels i and j in layer α based on the localized Gaussian distance as follows:

$$A_{\alpha i \alpha j} = \begin{cases} e^{-\frac{\|\mathbf{X}_{\alpha,i} - \mathbf{X}_{\alpha,j}\|_2^2}{\sigma^2}}, & \text{dis}_1(\mathbf{X}_{\alpha,i}, \mathbf{X}_{\alpha,j}) < p, \\ & \text{dis}_2(p(\alpha,i), p(\alpha,j)) < q; \\ 0, & \text{otherwise}. \end{cases} \tag{4}$$

where σ, p and q are design parameters and $p(\alpha, i)$ is the position of the superpixel i in layer α.

Beyond the traditional Gaussian distance, our intralayer connections consider two conditions for determining the presence of links: (1) features between two nodes should be similar; and (2) two connected superpixels should be in a localized region in the HSI. The first condition ensures the similarity of connected nodes while the second condition emphasizes geometric closeness in the HSI. For an initial setup of the parameters, we define $\text{dis}_1 = ||\mathbf{X}_{\alpha,i} - \mathbf{X}_{\alpha,j}||_2$, and define dis_2 as the Euclidean distance between the respective centroids of two superpixels. In terms of designing parameters, we set p as the mean of all pairwise similarities and tune the parameters q, σ based on the specific dataset. Details of parameter selection in practical applications can be found in Sect. 3.2.

2.1.3 Algebraic representations for HSI in M-GSP

Following the aforementioned process, we can easily construct HSI as a MLG with M layers and N nodes in each layer. In M-GSP [16], such multilayer graph structure can be intuitively represented by a fourth-order tensor $\mathbf{A} \in \mathbb{R}^{M\times N\times M\times N}$ defined as

$$\mathbf{A} = (A_{\alpha i\beta j}) \quad 1 \le \alpha, \beta \le M, 1 \le i, j \le N, \tag{5}$$

where each entry is calculated as Eqs. (3) and (4). Similar to normal graphs, Laplacian tensor $\mathbf{L} = \mathbf{D} - \mathbf{A} \in \mathbb{R}^{M\times N\times M\times N}$ can be defined as the alternative representation of MLG, where $\mathbf{D}$ is the degree tensor with the node degrees as its diagonal entries [16].

Discussion Before we venture into the M-GSP spectral representation of HSI, we provide a discussion on the interpretation of tensor representation and its relationship to the aforementioned "embedding" process in Sect. 2.1.2. Given a set of superpixels $\mathcal{X} = \{x_1, x_2, \cdots, x_N\}$, one can construct a vector $\mathbf{z}_i \in \mathbb{R}^N$ to characterize the underlying structural features of each superpixel i. Next, the interaction between two superpixels can be represented by a second-order tensor $\mathbf{A}_X = \sum_{i,j=1}^{N} a_{ij}\mathbf{z}_i \circ \mathbf{z}_j \in \mathbb{R}^{N\times N}$, where a_{ij} quantifies the relationship between superpixel i and j. Similarly, given a set of clustered bands (layers) $\mathcal{L} = \{l_1, l_2, \cdots, l_M\}$, a vector $\mathbf{z}_\alpha \in \mathbb{R}^M$ can capture the structural properties of layer α, and the connectivity between two layers could be represented by $\mathbf{A}_L = \sum_{\alpha,\beta=1}^{M} b_{\alpha\beta}\mathbf{z}_\alpha \circ \mathbf{z}_\beta \in \mathbb{R}^{M\times M}$. Following this approach, connectivity between the embedded nodes of superpixels in the layers can be represented by a fourth-order tensor to describe the features of superpixels and layers, i.e.,

$$\mathbf{A} = \sum_{\alpha,\beta=1}^{M}\sum_{i,j=1}^{N} w_{\alpha i\beta j}\mathbf{z}_\alpha \circ \mathbf{z}_i \circ \mathbf{z}_\beta \circ \mathbf{z}_j \in \mathbb{R}^{M\times N\times M\times N}, \tag{6}$$

where $\circ$ is the tensor outer product [45], $w_{\alpha i\beta j}$ is the weight of connection between the superpixel i's embedded node in layer α and the superpixel j's embedded node in layer β. More specially, if we select the vector $\mathbf{z}_i = [0, \cdots, 0, 1, 0, \cdots, 0]^{\mathrm{T}}$ in which the only nonzero element is the ith element (equal to 1) for both layers and superpixels, the fourth-order tensor becomes the adjacency tensor of the multilayer network. Interested readers may refer to [12, 16] for more details of the adjacency and Laplacian tensor.

2.1.4 Spectral representations for HSI in M-GSP

We now introduce the spectral representation of HSI in M-GSP. In M-GSP, multiple MLG spectra are introduced for different purposes. Since the MLG singular space is more robust when analyzing the order-wise features, we focus on the singular analysis in this work. More details of other MLG spectral analysis, e.g., M-GSP eigen-tensor analysis, can be found in [16]. With the adjacency tensor $\mathbf{A} \in \mathbb{R}^{M\times N\times M\times N}$, it can be decomposed via higher-order singular value decomposition (HOSVD) [46] as

$$\mathbf{A} = \mathbf{S} \times_1 \mathbf{U}^{(1)} \times_2 \mathbf{U}^{(2)} \times_3 \mathbf{U}^{(3)} \times_4 \mathbf{U}^{(4)}, \tag{7}$$

where $\times_n$ is the n-mode product [45] and $\mathbf{U}^{(n)} = [\mathbf{u}_1^{(n)} \quad \mathbf{u}_2^{(n)} \quad \cdots \quad \mathbf{u}_{I_n}^{(n)}]$ is a unitary $(I_n \times I_n)$ matrix, with $I_1 = I_3 = M$ and $I_2 = I_4 = N$. $\mathbf{S}$ is a $(I_1 \times I_2 \times I_3 \times I_4)$-tensor of which the subtensor $\mathbf{S}_{i_n}$ obtained by freezing the nth index to α:

- $< \mathbf{S}_{i_n=\alpha}, \mathbf{S}_{i_n=\beta} >= 0$ where $\alpha \neq \beta$.
- $||\mathbf{S}_{i_n=1}|| \geq ||\mathbf{S}_{i_n=2}|| \geq \cdots \geq ||\mathbf{S}_{i_n=I_n}|| \geq 0$.

The Frobenius-norms $\sigma_i^{(n)} = ||\mathbf{S}_{i_n=i}||$ is the n-mode singular value, with corresponding singular vectors in $\mathbf{U}^{(i)}$. Since the representing tensor shows partial symmetry in the undirected MLG, there are two modes of singular spectrum, i.e., $\mathbf{U}^{(1)} = \mathbf{U}^{(3)} = (\mathbf{f}_\alpha)$ characterizes the features of layers and $\mathbf{U}^{(2)} = \mathbf{U}^{(4)} = (\mathbf{e}_i)$ characterizes the nodes. Renotating $\mathbf{U}^{(1)} = \mathbf{U}^{(3)}$ as $\mathbf{F}_s = [\mathbf{f}_1 \cdots \mathbf{f}_M] \in \mathbb{R}^{M\times M}$ and $\mathbf{U}^{(2)} = \mathbf{U}^{(4)}$ as $\mathbf{E}_s = [\mathbf{e}_1 \cdots \mathbf{e}_N] \in \mathbb{R}^{N\times N}$, Eq. (7) can be written as

$$\mathbf{A} = \mathbf{S} \times_1 \mathbf{F}_s \times_2 \mathbf{E}_s \times_3 \mathbf{F}_s \times_4 \mathbf{E}_s. \tag{8}$$

With the singular tensors $\mathbf{F}_s$ and $\mathbf{E}_s$, the MLG singular transform (M-GST) for a MLG signal $\mathbf{s} \in \mathbb{R}^{M\times N}$ can be defined as

$$\check{\mathbf{s}} = \mathbf{F}_s^{\mathrm{T}} \mathbf{s} \mathbf{E}_s \in \mathbb{R}^{M\times N}. \tag{9}$$

Suppose that γ_i's are the layer-wise singular values and σ_i's are the node-wise singular values. The M-GSP spectral filter can be designed as

$$\mathbf{s}' = \mathbf{F}_s \begin{bmatrix} g(\gamma_1) & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & g(\gamma_N) \end{bmatrix} \mathbf{F}_s^{\mathrm{T}} \mathbf{s} \mathbf{E}_s \begin{bmatrix} f(\sigma_1) & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & f(\sigma_N) \end{bmatrix} \mathbf{E}_s^{\mathrm{T}}, \tag{10}$$

where functions $g(\cdot)$ and $f(\cdot)$ are designed by the specific tasks.

Here, we mainly focus on fundamentals of singular analysis of the undirected multilayer graphs. For more details on other concepts, such as MLG Fourier transform and M-GSP filter design, interested readers are referred to [16].

2.2 MLG-based unsupervised HSI segmentation

In this part, we propose an unsupervised segmentation approaches based on M-GSP spectral clustering.

Spectral clustering is an efficient method for unsupervised HSI segmentation [38]. Modeling HSI by a normal graph before spectral clustering, significant improvement is possible owing to its power in capturing the underlying structures [47]. However, by representing HSI by a single-layer graph, distinction of individual spectral bands might be overlooked. To capture the heterogeneous spectral–spatial structure in HSI, we propose to segment the HSI based on the M-GSP spectral clustering.

Given a HSI, we construct the multilayer graph as Sect. 2.1.2. We then calculate the spectra $\mathbf{F}_s = [\mathbf{f}_1 \cdots \mathbf{f}_M] \in \mathbb{R}^{M\times M}$ and $\mathbf{E}_s = [\mathbf{e}_1 \cdots \mathbf{e}_N] \in \mathbb{R}^{N\times N}$ to characterize the bands and superpixels, respectively, according to Eq. (8). Since we aim to segment superpixels into meaningful clusters, we focus on the superpixel-wise spectrum $\mathbf{E}_s$. Arranging $\mathbf{e}_i$ in the descending order of its corresponding singular value σ_i, i.e.,

$$\sigma_i = ||\mathbf{S}_{i_2=i}||, \tag{11}$$

where $\mathbf{S}_{i_2=i} \in \mathbb{R}^{M\times 1\times M\times N}$ is the subtensor of the core tensor $\mathbf{S}$ in Eq. (7) by freezing the second-order $i_2 = i$, we pick the first P singular vectors to preserve the most critical information for HSI based on the largest gap among the singular values. Clustering based on the P selected singular vectors and labeling each pixel within the superpixel, we can obtain a segmentation of the given HSI. The major process of MLG-based unsupervised segmentation is provided in Algorithm 1.

Algorithm 1 MLG-based Unsupervised HSI Segmentation (MLG-SC)

1: **Input**: HSI $\mathbf{I} \in \mathbb{R}^{K\times T}$ with K frames and T pixels, and the number of clusters Q;
2: **Multilayer Graph Construction**:
- Construct N superpixels for HSI based on ERS algorithm as $\mathbf{X} \in \mathbb{R}^{K\times N}$;
- Divide bands into M clusters based on k-means clustering;
- Construct an M-layer graph with adjacency tensor $\mathbf{A} \in \mathbb{R}^{M\times N\times M\times N}$ as Eq. (3) and Eq. (4);

3: **M-GSP Spectral Clustering**:
- Implement HOSVD on $\mathbf{A}$ to obtain the superpixel-wise singular tensors $\mathbf{E}_s = [\mathbf{e}_1 \cdots \mathbf{e}_N] \in \mathbb{R}^{N\times N}$ in the descending order of the 2-mode singular values σ_i as Eq. (11);
- Select the first P singular tensors as $\mathbf{P}_k = [\mathbf{e}_1, \cdots, \mathbf{e}_P] \in \mathbb{R}^{N\times P}$ based on the largest singular gap;
- Cluster the rows of $\mathbf{P}_k$ into Q groups based on k-means clustering;
- Cluster the superpixel i into group j if the i-th row of $\mathbf{P}_k$ is in group j;

4: **Segmentation**: Label the pixels as the same cluster of its superpixel;
5: **Output**: Segmented HSI.

2.2.1 Discussion

Before we dive further to develop MLG-based supervised HSI classification, we provide a short conceptual discussion on the M-GSP singular tensors. In literature, SVD is an efficient method to obtain the spectrum for signal analysis, such as spectral clustering and PCA analysis. In MLG-GSP, the order-wise singular vectors can be interpreted as subspaces characterizing features of frames and superpixels, respectively. Since HOSVD is robust and efficient, transforming signals to the MLG singular space (M-GST) for the analysis of underlying structures can be a useful alternative for M-GFT. More discussions on physical meaning of MLG spectrum can be found in [16]. To better understand the property of MLG-based singular tensors, we graphically illustrate the distribution of singular values compared to a graph-based model in Fig. 3. As shown, the energy of MLG-based singular values is more concentrated in the first few dominant singular

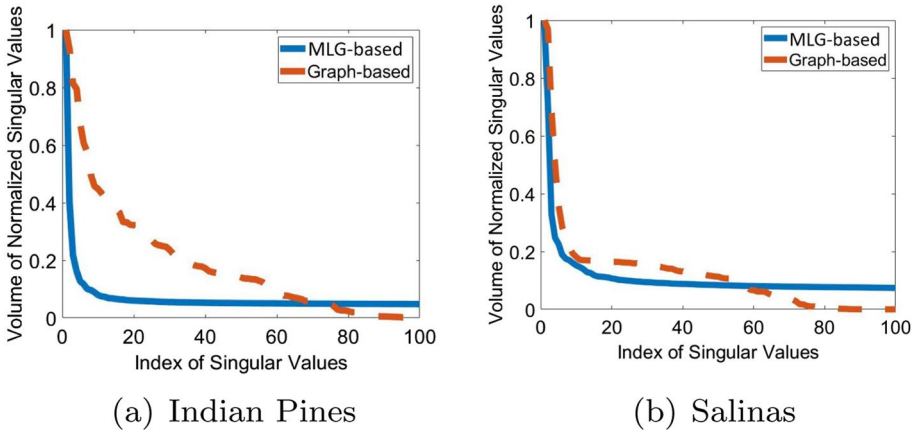


Fig. 3 Distribution of superpixel-wise singular values in HSI dataset

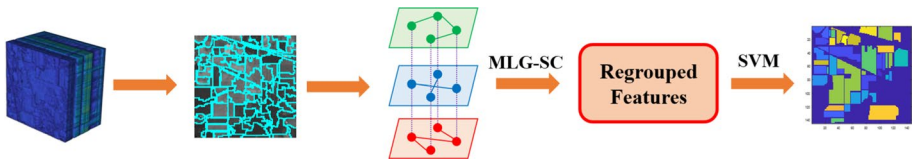


Fig. 4 Scheme of single-resolution segmentation

vectors in low frequency when compared against graph-based singular values. This energy concentration indicates a more convenient and low degradation implementation of spectral clustering within our proposed M-GSP framework.

2.3 MLG-based supervised HSI classification

In this part, we introduce the HSI classification based on M-GSP feature extraction.

2.3.1 Single-resolution of MLG-based HSI segmentation

We start with the single-resolution of the superpixels. In the superpixel-based classification, superpixel resolutions affect the final performance: finer resolution could capture more details, whereas coarse resolution captures the global information more efficiently. To benefit from both fine and coarse resolutions, we introduce the MLG-based spectral clustering on the fine resolution to regroup superpixels into a coarse resolution (the number of regrouped superpixels should still be larger than the number of classes) and use the regrouped features as classifier inputs. More specifically, we first implement MLG-based spectral clustering to cluster superpixels. We then combine the features of all pixels within the same cluster as the regrouped features. Finally, we update the features of each pixel by the regrouped features of its cluster and input the new features for classification. Here, we apply SVM to classify the regrouped features. The concept of our single-resolution HSI segmentation (MLG-SRC) is illustrated by Fig. 4, and the major steps are described in Algorithm 2.

Algorithm 2 Single-Resolution HSI Segmentation (MLG-SRC)

1: **Input**: HSI $\mathbf{I} \in \mathbb{R}^{K \times T}$ with K frames and T pixels;
2: **MLG Construction**:
- Construct N superpixels for HSI based on ERS algorithm as $\mathbf{X} \in \mathbb{R}^{K \times N}$;
- Divide bands into M clusters based on k-means clustering;
- Construct an M-layer graph with adjacency tensor $\mathbf{A} \in \mathbb{R}^{M \times N \times M \times N}$ as Eq. (3) and Eq. (4);

3: **Feature Regrouping**:
- Implement MLG-based spectral clustering to regroup the superpixels into D clusters, and combine the features of pixels within the same cluster as the regrouped features, i.e., $\mathbf{X}_R \in \mathbb{R}^{K \times D}$;
- Update the features of each pixel as the regrouped features of its cluster in $\mathbf{X}_R$ and generate the final input feature $\mathbf{X}_{MLG} \in \mathbb{R}^{K \times T}$;

4: **Classification**: Input $\mathbf{X}_{MLG}$ as features to SVM for classification;
5: **Output**: Segmented HSI.

The benefits of the proposed MLG-SRC include:

- Against a single resolution of coarse superpixels, the MLG-SRC implements an analysis step over a fine resolution and is capable of capturing detailed features. Against a single resolution of fine superpixels, the MLG-SRC substantially reduces pixel number and enhances robustness of the feature inputs to the classifier. Too many superpixels may make the features less distinctive and over-segment the regions, whereas too few superpixels may lead to boundary ambiguity.
- Traditional graph-based superpixel segmentation only captures a single-layer structure. M-GSP regrouping could reveal additional feature information across the heterogeneous multi-band structures.
- In traditional superpixel segmentation, the distinct regions are usually labeled as different superpixels. However, in MLG-SRC, superpixels from different regions may be grouped to the same cluster. Thus, regrouped features can involve similar pixels that cover a large distance and potentially generate more features.
- MLG-SRC can be easily integrated with other feature extraction or selection algorithms. Dimension reduction techniques such as PCA and ICA can potentially improve the performance when applied on features and feature groups generated by MLG-SRC.

2.3.2 Multi-resolution of MLG-based HSI segmentation

1) Multi-resolution structure:

Although MLG-based spectral clustering can regroup small superpixels into larger ones and to benefit from both fine and coarse resolutions in MLG-SRC, the initial resolution setting of superpixels still affects final performance (Fig. 5). As Fig. 6 shows, different initial resolutions can lead to different levels of accuracy. It is practically difficult to determine the optimal initial number of superpixels.

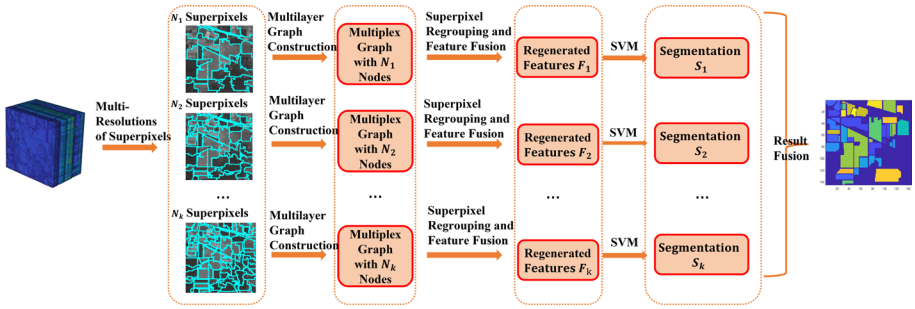


Fig. 5 Scheme of MLG-based supervised HSI classification

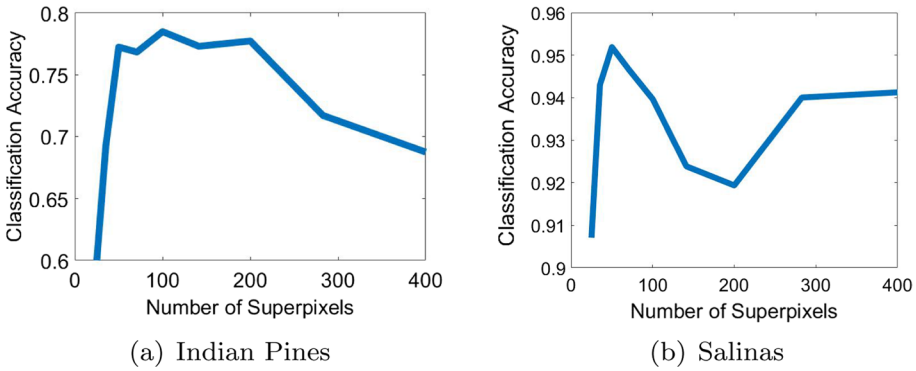


Fig. 6 Accuracy of MLG-SRC over different initial resolutions

Algorithm 3 Multi-Resolution HSI Segmentation (MLG-MRC)

1: **Input**: HSI $\mathbf{I} \in \mathbb{R}^{K \times T}$ with K frames and T pixels;
2: Construct multiple resolutions of superpixels for HSI;
3: Construct different multilayer graphs for each resolution of HSI;
4: Implement MLG-SRC for each resolution to obtain the sub-results;
5: Fuse all the sub-results for the final segmentation;
6: **Output**: Segmented HSI.

Here, we consider a multi-resolution structure of classification (MLG-MRC) shown as Fig. 5. In this framework, we examine several different initial resolutions of superpixels. MLG-SRC is applied to each initial resolution to regroup the superpixels with a same reduction ratio in group numbers, i.e., 70% of initial superpixels. Applying SVM to classifying the multiple regrouped superpixels, we fuse the results from different initial resolutions in final segmentation. The algorithm is described in Algorithm 3. Although the multi-resolution structures have been considered in literature, MLG-MRC exhibits two major distinctions. First, we apply a novel MLG-based clustering algorithm to regroup the superpixels and generate new features for classification. Second, we provide several novel decision fusion strategies, based on both confidence score and graph structures to be discussed below.

2) Decision fusion:

Majority voting (MV) [48] is a widely used fusion scheme for different resolutions. In this method, the label l of a specific pixel is determined by

$$l = \arg\max_i \sum_{j=1}^{C} w_j \cdot \delta(l_j), \tag{12}$$

where C is the number of distinct resolutions, l_j is the label of the pixel in resolution j, w_j is the voting strength, and $\delta(l_j) = 1$ if $l_j = i$; otherwise, $\delta(l_j) = 0$. Note that a basic majority voting based on equal strength $w_j = C^{-1}$ applies the same strength to different resolutions which would ignore their differences. To improve decision fusion, we introduce several novel strategies for the decision fusion.

- *Validation Accuracy* (VA): One intuitive way to design decision strength is based on validation accuracy. If the validation accuracy of a certain resolution is larger, it might also have better performance in the test data. Thus, we can assign larger weights to the resolution with better validation accuracy. Here, we can apply the validation accuracy directly as the weighting strength w_j for resolution j to fuse the decision according to Eq. (12).
- *Decision Value* (DV): As one alternative, the decision probability for each class of pixels can be used as the weight. In multi-class SVM, the predicted label is determined according to the decision value $\mathbf{p} \in \mathbb{R}^C$, where C is the number of classes [49]. If the decision value is larger, it means that SVM has higher classification confidence. Thus, decision values can also describe the confidence levels of classification results. Let $\mathbf{p}_{ij}$ be the decision value of pixel i in jth resolution. We set the weight of l_{ij} to

  $$w_{ij} = \max_k \mathbf{p}_{ij}(k). \tag{13}$$

  Unlike validation accuracy which is the same for all pixels in each resolution, this weight based on decision value may vary even for pixels at the same resolution.
- *Graph Total Variation* (TV): Graph-based metrics can serve as weights. For a robust setup of superpixels, signals should be smooth and exhibit stable underlying graph structure. To this end, we introduce graph-based total variation to measure smoothness. Given a superpixel segmentation j of a HSI with N superpixels and K spectral frames, we regenerate the features of each superpixel by averaging all pixels within. We then construct a single-layer graph based on Gaussian distance to measure similarity between different superpixels. Defining a Laplacian matrix by $\mathbf{L} = \mathbf{D} - \mathbf{A}$ where $\mathbf{D}$ is the degree matrix and $\mathbf{A}$ is the adjacency matrix, the total variation [7] of the feature signal $\mathbf{s}_p \in \mathbb{R}^N$ for the pth band frame over $\mathbf{L}$ is

  $$\mathrm{TV}_p = ||\mathbf{s}_p - \frac{1}{|\lambda_{\max}|}\mathbf{L}\mathbf{s}_p||_2^2, \tag{14}$$

  where $\lambda_{\max}$ is the largest eigenvalue of $\mathbf{L}$. Total variation describes the propagation differences between two steps. A smaller total variation indicates a more smooth signal. With K frames in total, final smoothness for resolution j is defined as $\mathrm{SM}_j = \frac{1}{K}\sum_p \mathrm{TV}_p$. Since we prefer a larger weight for the smooth signal, the final weight of the resolution j is defined as

  $$w_j = e^{-SM_j}. \tag{15}$$

- *Von Neumann Entropy* (VN): The stability of the underlying graph structure can also indicate the confidence level of a specific superpixel resolution. In quantum theory [50], a pure state leads to a zero Von Neumann entropy. The entropy is larger if there are more mixed states in the system. Similarly, in our HSI analysis, since we prefer a stable system or a pure state on the underlying graph, the weight should be larger if the Von Neumann entropy is smaller. Here, we introduce the Von Neumann entropy to evaluate the graph stability [50]. Similar to total variation, a Laplacian matrix $\mathbf{L}$ can be defined with adjacency matrix $\mathbf{A} = (a_{pq})$ for the jth resolution. First, define $c = 1/(\sum_{p,q} a_{pq})$ and rescale the Laplacian matrix

$$\mathbf{L}_G = c \cdot (\mathbf{D} - \mathbf{A}), \tag{16}$$

We can define the weight for the jth resolution as

$$w_j = e^{-h_j}. \tag{17}$$

based on the Von Neumann entropy

$$h_j = -\mathrm{Tr}[\mathbf{L}_G \log_2 \mathbf{L}_G]. \tag{18}$$

Note that, here we provided several possible alternatives for the weights of decision fusion. The performances of the various proposed fusion strategies will be presented in Sect. 3.

3 Results and discussions

We now test the performance of the proposed unsupervised segmentation and supervised HSI classification approaches in several well-known datasets to demonstrate the efficacy of M-GSP in HSI analysis. We also comparatively test the performance of various different fusion decisions.

3.1 Dataset

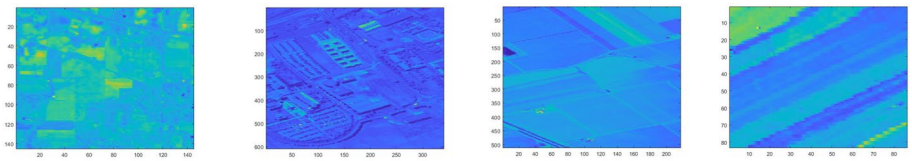
We test the performances of the proposed methods based on four public HSI datasets accessible from website[1]. The first HSI is *Indian Pines* (IndianP) scene originally gathered by AVIRIS sensors over an agricultural field. The second HSI dataset is the *University of Pavia* (PaviaU) acquired by ROSIS sensor. Note that some of the samples in PaviaU contain no information and have to be discarded before analysis. Two other HSIs used in the experiments are the *Salinas Scene* (Salinas) and *Salinas-A Scene* (SalinasA) datasets, which were collected by the 224-band AVIRIS sensor over Salinas Valley, California, and exhibit high spatial resolution. For each dataset, we have groundtruth classes for part of samples.

For these HSIs, Table 1 provides vital statistics and we provide visual illustration of the geometric plots in Figs. 7 and 8. Note that, Fig. 8 treats the unlabeled groundtruth samples as backgrounds with the same class label. Interested readers can find more information on the HSI datasets at the website[1].

[1] http://www.ehu.eus/ccwintco/index.php/Hyperspectral_Remote_Sensing_Scenes

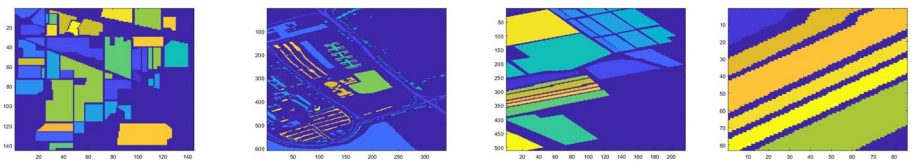
Table 1 Statistics of different HSI datasets

HSI	Pixel size	# of bands	# of classes	# of labeled samples (percentage)	Average of pixel # per class	Std of pixel # per class
IndianP	145 × 145	200	16	10249 (48.75)	640.6	650.15
PaviaU	610 × 340	103	9	42776 (20.62)	4752.9	5540.3
Salians	512 × 217	204	16	54129 (48.72)	3383.1	2774.7
SaliansA	83 × 86	204	9	5348 (74.92)	891.3	444.4



(a) Indian Pines (b) Pavia University (c) Salinas (d) SalinasA

Fig. 7 Mean of HSI over spectral dimension



(a) Indian Pines (b) Pavia University (c) Salinas (d) SalinasA

Fig. 8 Ground truth of class labels

3.2 Unsupervised HSI segmentation

In this part, we first test the performance of unsupervised HSI segmentation. Although our proposed MLG-based method is amenable to various sophisticated clustering approaches, we find it easier to demonstrate it by using a basic spectral clustering scheme. Here, we mainly use comparison of basic methods, such as *k*-means clustering and GSP-based spectral clustering, to demonstrate the power of M-GSP in processing the HSI datasets.

To validate the performance of different methods, we consider two experimental setups. In the first scenario, we carry out unsupervised segmentation on all data samples and evaluate the overall visualization results of labeled samples. In the second stage, we process all data samples but focus on detecting boundaries (edges) of each cluster, in terms of both visualization results and numerical accuracy.

For fair comparison, we segment the HSI into N superpixels first before applying respective clustering algorithms thereupon. For the GSP-based method, we construct the graph $\mathbf{W} \in \mathbb{R}^{N\times N}$ using Gaussian distance

$$W_{ij} = \begin{cases} e^{-\frac{||\mathbf{s}_i - \mathbf{s}_j||_2^2}{\sigma^2}}, & \left\|\mathbf{s}_i - \mathbf{s}_j\right\|_2^2 \leq \tau \\ 0, & \text{otherwise} \end{cases} \tag{19}$$

where $\mathbf{s}_i$ denotes the feature of ith superpixel. The threshold τ is set to the statistical mean of all pairwise distances among superpixels, and σ is tunable according to specific datasets. For M-GSP-based methods, we construct the multilayer graph with $M = 10$

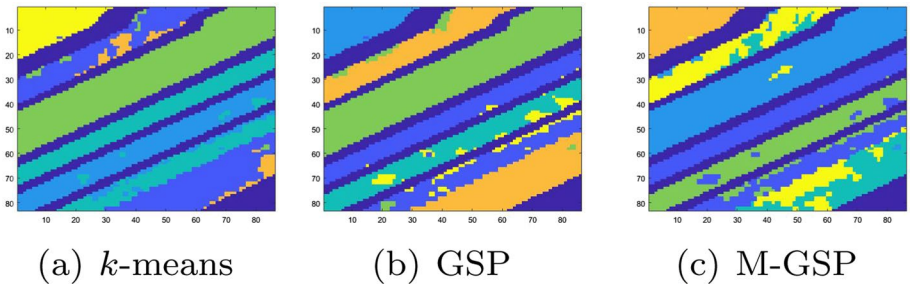


Fig. 9 Segmented results of Salinas A

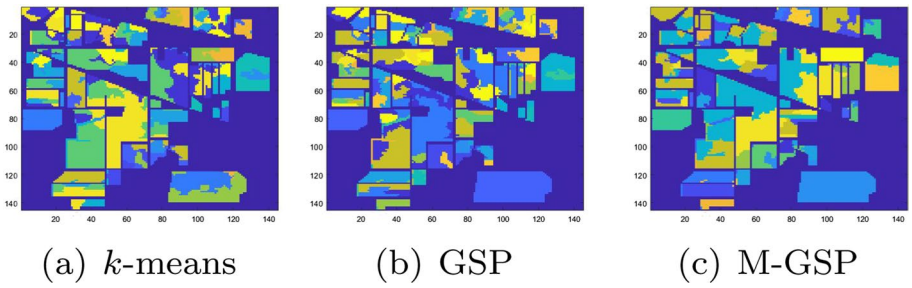


Fig. 10 Segmented results of Indian Pines

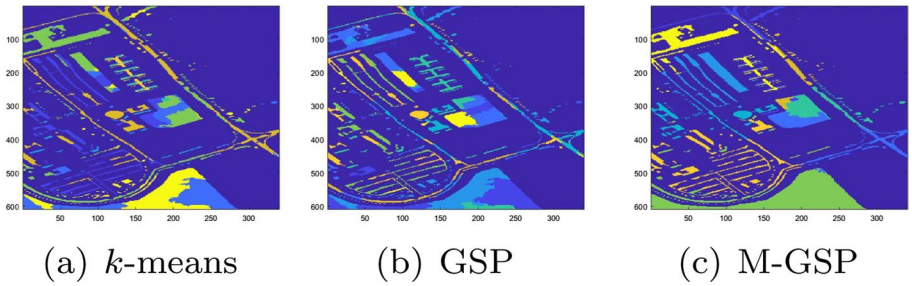
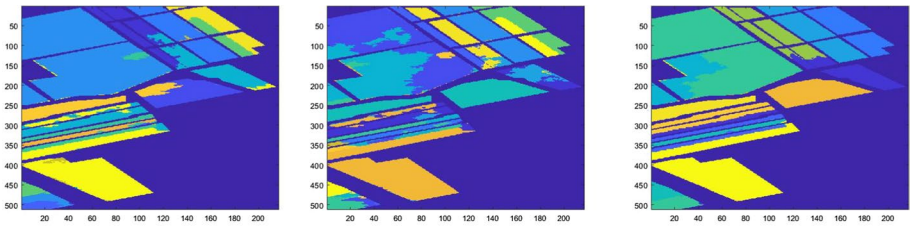


Fig. 11 Segmented results of Pavia University

layers, and calculate the distance based on Eqs. (3) and (4). The parameter p in Eq. (4) is also set to the statistical mean of all pairwise intralayer feature distances, and $q = 100$ is used. We select the number of key spectra based on the largest gap of singular values. We summarize the results below.

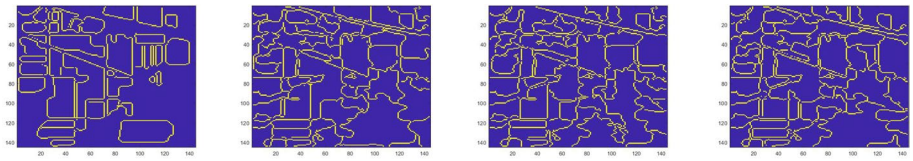
3.2.1 Visualization of HSI segmentation

By setting $N = 500$ for all tested HSIs, Figs. 9, 10, 11, 12 present the visualization results of the three HSI segmentations using different algorithm in comparison to the groundtruth data. These results show that, in general, the MLG-based spectral clustering (M-GSP) displays more stable segmentations than the single layer GSP-based spectral clustering (GSP) as well as the k-means algorithm. Since it is harder to evaluate the details for too many classes, we can focus more on the exemplary *Salinas A* dataset. In particular, Fig. 8d shows six different groundtruth classes. However, both k-means



(a) k-means (b) GSP (c) M-GSP

Fig. 12 Segmented results of Salinas



(a) Ground Truth (b) k-means (c) GSP (d) M-GSP

Fig. 13 Boundaries in Indian Pines

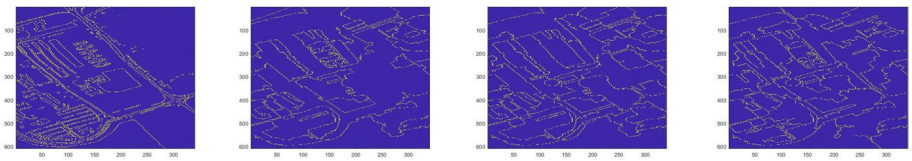
and GSP-based method failed to detect the class No.6, marked by orange in Fig. 9a and marked by yellow in Fig. 9b. Meanwhile, in Fig. 9c, the M-GSP method successfully identified all six classes and delivered results that are closer to groundtruth. Recall that, from Fig. 3, the MLG-based singular values are more concentrated, which provides the benefits of robustness in spectral clustering. Results from "Indian Pines" and "Pavia University" also show similarly stronger performance for M-GSP. These results collectively demonstrate improved efficiency of M-GSP in unsupervised HSI segmentation.

3.2.2 Boundary segmentation

Although unsupervised methods may generate meaningful segmentation different from groundtruth, we are still interested in how far the segmented results are from the true labels. Since it is inefficient to match all the clusters to the corresponding true labels, we focus on the boundaries of the segmentation. In the boundary detection, we set $N = 100$ and define the accuracy as

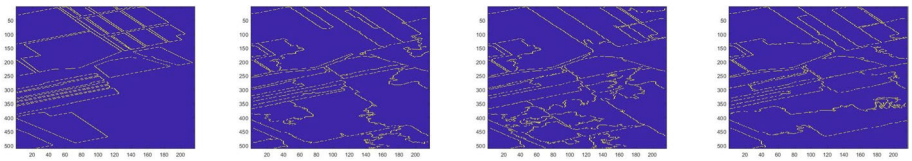
$$\text{Acc} = \sum_{i=1}^{T} 1(L_i = \hat{L}_i)/T, \tag{20}$$

where T is the number of pixels in the HSI, L_i is the true labels of edges for the ith pixel, $\hat{L}_i$ is the estimated labels, and $1(\cdot)$ denotes the indicator function. The boundary results are visualized in Figs. 13, 14, 15. Since we consider the unlabeled samples in the clustering process, there can be more details for the HSIs than their corresponding groundtruth, especially for Pavia University and Indian Pines. However, we can still derive some benefits of M-GSP in boundary detection. As Fig. 15 shows, M-GSP can generate clearer edges, obviously on the top half image, while GSP and k-means appear to over-segment. We also present the accuracy defined in Eq. (20) as Table 2. These results show that M-GSP performs better than k-means and GSP. They demonstrate the



(a) Ground Truth (b) k-means (c) GSP (d) M-GSP

Fig. 14 Boundaries in Pavia University



(a) Ground Truth (b) k-means (c) GSP (d) M-GSP

Fig. 15 Boundaries in Salinas

Table 2 Accuracy of segmentation boundaries

Data	*k*-means	GSP	M-GSP
IndianP	0.8257	0.8298	**0.8441**
Salinas	0.9208	0.9285	**0.9409**
PaviaU	0.9070	0.9088	**0.9255**

The best performances are marked in bold

efficiency of MLG models in HSI analysis. Note that we do not claim M-GSP to be the best approaches for all scenarios. Without impractically requiring hindsight to fine-tune various parameters for each HSI dataset to generate "the best results," M-GSP delivers consistently strong and stable segmentation results for various HSI datasets by relying on some basic guidelines for selecting parameters.

3.3 Supervised HSI classification

We next test M-GSP in supervised HSI classification.

3.3.1 Overall accuracy

Applying M-GSP-based spectral clustering as feature extraction of HSI, we compare the proposed algorithms with several well-known feature extraction algorithms, including NPE [51], LP-NPE [52], LDA [53], LFDA [54], SPCA [39] and MSPCA [39]. For the proposed MLG-SRC and MLG-MRC, we regroup the superpixels to 70% of the original superpixel number, before extracting features based on M-GSP. Here, we show the results of MLG-MRC according to different decision values (fusion weights). More analysis of different fusion strategies will be illustrated further in Sect. 3.3.2.

The overall accuracy under different numbers of training samples per class (TS/C) is shown in Table 3. In this experiment, parameters of multiple resolutions are tuned for

Table 3 Overall accuracy of different HSI segmentation

Data	TS/C	NPE	LPNPE	LDA	LFDA	SPCA	MSPCA	MLG-SRC	MLG-MRC
IndianP	5	0.5368	0.6725	0.5995	0.5962	0.7734	0.7868	0.7432	**0.7936**
	10	0.7049	0.7645	0.6930	0.6491	0.8576	0.8712	0.8604	**0.8773**
	20	0.7987	0.8351	0.7656	0.7401	0.9390	**0.9569**	0.9226	0.9545
PaviaU	5	0.6835	0.7612	0.7243	0.7467	0.7439	0.7849	0.7561	**0.8236**
	10	0.8063	0.8255	0.8124	0.7895	0.8342	**0.9167**	0.8398	0.8896
	20	0.8569	0.8856	0.8500	0.8698	0.8938	**0.9537**	0.8697	0.9432
Salinas	5	0.8486	0.9209	0.8903	0.8883	0.9442	0.9500	0.9499	**0.9588**
	10	0.8899	0.9452	0.9146	0.8277	0.9678	0.9815	0.9614	**0.9863**
	20	0.9069	0.9589	0.9372	0.9356	0.9837	0.9904	0.9840	**0.9915**

The best performances are marked in bold

Table 4 Performance of different fusion strategies

	TS/C	MV	VA	DV	VN	TV
Indian Pines						
MS- PCA	5	0.6663	0.7241	**0.7334**	0.7263	0.7265
	10	0.7476	0.7778	**0.7829**	0.7661	0.7661
MLG- MRC	5	0.7164	**0.7338**	0.7260	0.7271	0.7295
	10	0.8220	0.8253	**0.8253**	0.7886	0.7823
Pavia University						
MS- PCA	5	0.7078	0.7440	0.7317	**0.7798**	0.7712
	10	0.8447	0.8541	0.8553	**0.8588**	0.8500
MLG- MRC	5	0.8162	0.8140	**0.8240**	0.7977	0.8085
	10	0.8549	0.8586	**0.8605**	0.8459	0.8435
Salinas						
MS- PCA	5	0.8521	0.9256	0.9349	**0.9655**	0.9634
	10	0.9662	0.9774	0.9718	**0.9789**	0.9765
MLG- MRC	5	0.9469	0.9456	0.9485	**0.9844**	0.9622
	10	0.9872	0.9874	0.9873	0.9857	**0.9942**

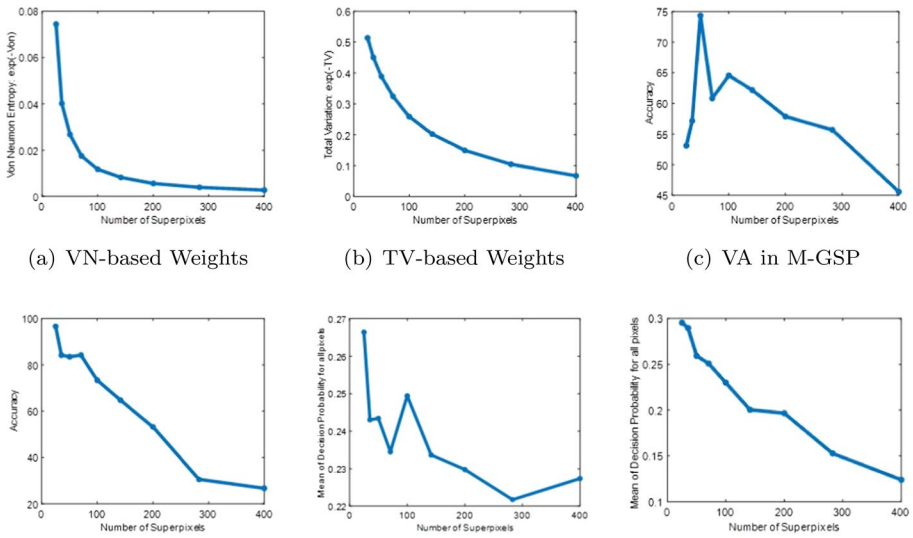
The best performances are marked in bold

different HSIs. From the test results, the proposed MLG-MRC exhibits a superior overall performance, especially for those scenarios with fewer classes.

3.3.2 Analysis of different fusion strategies

Now, we analyze the performance of different fusion strategies. Since MSPCA [39] also fuses multiple resolutions of superpixels, we investigate our proposed fusion methods in combination with both MSPCA and MLG-MRC.

We fuse the results from 9 resolutions, i.e., $N_i \in [25, 35, 50, 70, 100, 140, 200, 280, 400]$, for all comparative methods. The test accuracy is shown in Table 4. As shown, our proposed weight approaches lead to significant improvement over the basic majority voting (VT) for MSPCA. Compared to SVM-based weights, the graph-based weights are better for MSPCA since additional geometric information is considered. For MLG-MRC, the proposed weights show a slight improvement, which suggests that MLG-MRC is less sensitive to different decision strengths. To better understand the effect of different



(a) VN-based Weights (b) TV-based Weights (c) VA in M-GSP

(d) VA in MSPCA (e) Mean DV in M-GSP (f) Mean DV in MSPCA

Fig. 16 Decision strengths over different numbers of superpixels

weights, we also present decision strengths over different numbers of superpixels in Fig. 16. In Fig. 16a, b, the graph-based weights favor low resolution to form baselines and use high resolutions to interpolate details. Thus, for HSIs with larger area of segmented groups, such as Salinas, the graph-based weights generate superior performances. Using SVM-based weights, we find no consistent trend among different superpixels, and the results vary for different datasets. Since the MLG-based methods have already incorporated the underlying geometric structures, they continue to display robust results even when using SVM-based weights.

3.3.3 Robustness in noisy datasets

We further evaluate the robustness of proposed methods in noisy environment. More specifically, we consider two types of noise models: (1) pixel-dependent noise where pixel noise variance depending on corresponding pixel data value; and (2) non-pixel dependent noise where noise variance is defined by mean of all pixel values. These two different noise models describe two different practical sensing noises. We also test both uniform noise and Gaussian noise. From the test performances shown in Table 5, we find the newly proposed MLG-based methods to be less sensitive to various types of sample noises.

3.3.4 Complexity

In practice, the M-GSP analysis usually consists of two parts: 1) calculation of the MLG spectra; and 2) implementation of algorithms with the calculated spectra for specific tasks. In this section, we evaluate the complexity of M-GSP analysis compared with traditional graph-based methods and signal processing approaches.

- *Graph Construction and Spectrum Calculation*: We compare the complexity of spectra calculation for M-GSP with several typical graph models for multilayer feature

Table 5 Overall accuracy in noisy environment

(Uniform/Gaussian)		Indian Pines		Salinas	
TS/C	Noise level	MPCA	MLG-MRC	MSPCA	MLG-MRC
Setup 1					
5	5%	0.7140/0.7012	**0.7377/0.7493**	0.8592/0.8670	**0.9231/0.9159**
	10%	0.7110/0.6702	**0.7135/0.7276**	0.8341/0.8421	**0.8956/0.9023**
	15%	0.6868/0.6507	**0.7133/0.7215**	0.8107/0.8215	**0.8815/0.8975**
5 10	5%	0.7613/0.7753	**0.8033/0.8056**	0.9509/0.9647	**0.9632/0.9684**
	10%	0.7522/0.7498	**0.7788/0.7984**	0.9408/0.9431	**0.9445/0.9491**
	15%	0.7317/0.7382	**0.7544/0.7845**	0.9381/**0.9368**	**0.9421**/0.9317
Setup 2					
5 5	5%	0.6839/0.6740	**0.7023/0.6898**	0.8591/0.8679	**0.9240/0.9050**
	10%	0.6640/0.6501	**0.6764/0.6647**	0.8471/0.8512	**0.9045/0.8878**
	15%	0.6413/0.6247	**0.6687/0.6427**	0.8317/0.8421	**0.8915/0.8775**
5 10	5%	0.7579/0.7408	**0.8022/0.7810**	0.9682/0.9621	**0.9735/0.9633**
	10%	0.7409/0.7333	**0.7800/0.7762**	0.9521/0.9450	**0.9547/0.9532**
	15%	0.7208/0.7235	**0.7695/0.7584**	**0.9437/0.9416**	0.9305/0.9241

The best performances are marked in bold

extraction. Suppose we have an HSI with M clustered bands and N superpixels. The first step is to calculate the graph spectra for interlayer connections (cross-band) and intralayer connections (spatial), respectively. For the single-layer graph model, one method (2-Way GSP [15]) is to consider all spectrum bands with the same graph structure, and each superpixel has the same cross-band interactions. Note that the bands are treated as nodes for cross-band interactions in this model. Now, the HSI can be represented by one $M \times M$ matrix for cross-band interactions and one $N \times N$ matrix for spatial connections. Eigen-decomposition is then implemented for these two matrices, respectively. However, such model assumes the same graph structure for all bands which is unable to represent different spatial connections for different bands. To compensate, one can also construct individual graphs for each band (Individual Graphs). If one also considers the same cross-band connections as 2-way GSP, the HSI can be represented by M $N \times N$ matrices for spatial connections and one $M \times M$ matrix for cross-band interactions. Moreover, if different cross-band connections are considered for different superpixels (Complex Individual Graphs), the HSI could be represented by M $N \times N$ matrices for spatial connections and N $M \times M$ matrices for cross-band connections. Here, we only provide comparison with several typical models, leaving further comparison to future works. In M-GSP, we allow different intra-layer connections for each layer, and flexible interlayer interactions. No matter how the interactions are defined, HSI is always represented by one $M \times N \times M \times N$ tensor. The structures of interactions are represented by the entry values, and the identification of layers/nodes is reflected by the index of each entry in the tensor. We then use HOSVD to obtain all the spectra. To compare the complexity, we test different models in the *Pavia University* dataset and set $M = 10$. We test the model by using Matlab programs on a desktop (Intel

Table 6 Running time of graph construction + spectra calculation (Seconds)

	# of matrices/ tensors	N=100	N=500	N=1000
M-GSP	1	0.082	2.319	13.94
2-Way GSP	2	0.007	0.122	0.509
Individual Graphs (GSP)	M+1	0.117	2.405	11.62
Complex Individual Graphs (GSP)	M+N	0.118	2.412	12.57

Table 7 Performance under different regrouping ratio

Regrouping ratio		0.8	0.7	0.6	0.5	0.4	0.3	0.2
MLG-SRC		0.7107	0.7806	0.7857	0.7275	0.6427	0.6831	0.5716
MLG-MRC		0.8018	0.8419	0.8044	0.7916	0.7475	0.5522	0.4218
SPCA	0.6985							
MPCA	0.6990							

Xeon CPU @ 3.50GHz, 32 GB RAM), where eigen-decomposition is implemented for single-layer graphs, and HOSVD in the tensor toolbox[2] is used for M-GSP. The runtime of spectra calculation with graph construction under different N is shown as Table 6. Since 2-way GSP neglects the different structures for different layers and superpixels, it requires the least complexity but provides less information. It is unable to represent the heterogeneous graph structures for different bands. If different interlayer and intralayer connections are considered, M-GSP has similar or slightly better complexity than graph-based models. Moreover, we only need to use one tensor to represent the HSI, which indicates that M-GSP provides a more general representation for different HSIs and MLGs.

- *Implementation of M-GSP Algorithms*: Since our proposed M-GSP processing can be flexibly integrated with other dimension reduction methods to reduce complexity, we find it unnecessary to provide evaluation of computation complexity for various setups. In general, the original MLG-MRC has a similar runtime as MSPCA. For example, under the same settings of multi-resolutions as Sect. 3.3.1 in Indian Pines HSI with tuning parameters of SVM among 15 sets, the runtimes for MLG-MRC and MSPCA are 28.21 seconds and 25.37 seconds, respectively. These and other tests indicate similar computation complexity for methods based on M-GSP and PCA.

3.3.5 Impact of regrouping ratio

To provide a better interpretation of M-GSP regrouping, we measure the performance under different regrouping ratio. We compare with SPCA and MPCA as an example, where $\mathrm{TS}/C = 5$. The majority voting is applied for multi-resolution methods. We do not regroup the features of SPCA and MPCA via M-GSP. The results are shown as Table. 7. From the results, we proposed methods have better performance when the ratio is above 30%. When the ratio is too small, some resolutions may have few clusters which

[2] https://www.tensortoolbox.org/

reduce the accuracy of classification. In general, the ratio can be set around 60% to achieve a robust performance.

4 Conclusions

This work introduces M-GSP in hyperspectral imaging processing. To capture heterogeneous underlying structures within different but highly correlated spectrum frames in hyperspectral images (HSIs), we propose to represent HSIs via multilayer graphs. Analyzing singular spectra of adjacency tensor for the multilayer graph, we first develop a MLG-based spectral clustering for unsupervised HSI segmentation. Leveraging features extracted with M-GSP, we propose two algorithms for supervised HSI classification. We also consider several novel decision fusion strategies for multiple resolution superpixel analysis. Our experimental results demonstrate the robustness and efficiency of the proposed methods, successfully showcasing the power of M-GSP in HSI analysis.

Future direction Advances in tensor algebra and multilayer graph theory present more opportunities to explore M-GSP and its applications in hyperspectral imaging. One interesting topic is the construction of multilayer graphs efficiently for HSIs. In this work, we provide an easy-to-implement multilayer graph construction based on Gaussian distance, to explore the power of M-GSP. More efficient multilayer graph construction could lead to further performance gains. Potential approaches include those based on feature similarities and graph spectral properties [55]. Another promising direction is the use of M-GSP transformation and spectral filters in HSI analysis. More analytical tools, such as multilayer graph filter banks and sampling theory, can be introduced for HSI processing, including the extraction of HSI's underlying spatial-frequency relationship. Other interesting future directions include the development of MLG convolutional networks and the integration of M-GSP with deep neural networks.

Abbreviations

HSI	Hyperspectral images
GSP	Graph signal processing
M-GSP	Multilayer graph signal processing
MLG	Multilayer graphs/networks
HOSVD	Higher-order singular value decomposition
SVM	Support vector machine.

Acknowledgements
Not applicable.

Author Contributions
SZ developed the basic approaches and wrote the initial manuscript draft. Other authors assisted in improving the framework and revised the paper. All readers read and approved the final manuscript.

Funding
This material is based upon work supported by the National Science Foundation under Grant No. 1824553, Grant No. 2029027, and Grant No. 2029848.

Availability of data and materials
We test the performances of the proposed methods based on four public HSI datasets accessible from website http://www.ehu.eus/ccwintco/index.php/Hyperspectral_Remote_Sensing_Scenes.

Declarations

Ethics approval and consent to participate
Not applicable.

Consent for publication
Not applicable.

Competing interests
The authors declare that they have no competing interests.

Received: 21 April 2022 Accepted: 21 September 2022
Published online: 01 October 2022

References

1. W.-K. Ma, J.M. Bioucas-Dias, T.-H. Chan, N. Gillis, P. Gader, A.J. Plaza, A. Ambikapathi, C.-Y. Chi, A signal processing perspective on hyperspectral unmixing: Insights from remote sensing. IEEE Signal Process. Mag. **31**(1), 67–81 (2014)
2. M. Fauvel, Y. Tarabalka, J.A. Benediktsson, J. Chanussot, J.C. Tilton, Advances in spectral-spatial classification of hyperspectral images. Proc. IEEE **101**(3), 652–675 (2013)
3. L. Zhang, L. Zhang, B. Du, Deep learning for remote sensing data: A technical tutorial on the state of the art. IEEE Geosci. Remote Sens. Mag. **4**(2), 22–40 (2016)
4. W. Zhao, S. Du, Spectral-spatial feature extraction for hyperspectral image classification: A dimension reduction and deep learning approach. IEEE Trans. Geosci. Remote Sens. **54**(8), 4544–4554 (2016)
5. Y.-Z. Feng, D.-W. Sun, Application of hyperspectral imaging in food safety inspection and control: a review. Crit. Rev. Food Sci. Nutr. **52**(11), 1039–1058 (2012)
6. H.Z. Shafri, E. Taherzadeh, S. Mansor, R. Ashurov, Hyperspectral remote sensing of urban areas: an overview of techniques and applications. Res. J. Appl. Sci. Eng. Technol. **4**(11), 1557–1565 (2012)
7. A. Ortega, P. Frossard, J. Kovačević, J.M.F. Moura, P. Vandergheynst, Graph signal processing: Overview, challenges, and applications. Proc. IEEE **106**(5), 808–828 (2018). https://doi.org/10.1109/JPROC.2018.2820126
8. T.N. Kipf, M. Welling, Semi-supervised classification with graph convolutional networks. In: International Conference on Learning Representations (ICLR) (2017)
9. J. Zeng, G. Cheung, Y.-H. Chao, I. J. Serra-Sagristà, A. Ortega, Hyperspectral image coding using graph wavelets. In: 2017 IEEE International Conference on Image Processing (ICIP), pp. 1672–1676 (2017)
10. D. Hong, L. Gao, J. Yao, B. Zhang, A. Plaza, J. Chanussot, Graph convolutional networks for hyperspectral image classification. IEEE Trans. Geosci. Remote Sens. **59**(7), 5966–5978 (2021)
11. J. Wang, Y. Zhang, X. Yuan, Z. Meng, Z. Tao, Calibrated hyperspectral image reconstruction via graph-based self-tuning network. arXiv preprint arXiv:2112.15362 (2021)
12. M. De Domenico, A. Solé-Ribalta, E. Cozzo, M. Kivelä, Y. Moreno, M.A. Porter, S. Gómez, A. Arenas, Mathematical formulation of multilayer networks. Phys. Rev. X **3**(4), 041022 (2013)
13. S. Zhang, H. Zhang, H. Li, S. Cui, Tensor-based spectral analysis of cascading failures over multilayer complex systems. In: 2018 56th Annual Allerton Conference on Communication, Control, and Computing (Allerton), pp. 997–1004 (2018). IEEE
14. F. Grassi, A. Loukas, N. Perraudin, B. Ricaud, A time-vertex signal processing framework: Scalable processing and meaningful representations for time-series on graphs. IEEE Trans. Signal Process. **66**(3), 817–829 (2018). https://doi.org/10.1109/TSP.2017.2775589
15. P. Das, A. Ortega, Graph-based skeleton data compression. In: 2020 IEEE 22nd International Workshop on Multimedia Signal Processing (MMSP), pp. 1–6 (2020). https://doi.org/10.1109/MMSP48831.2020.9287103
16. S. Zhang, Q. Deng, Z. Ding, Introducing graph signal processing over multilayer networks: Theoretical foundations and frequency analysis. arXiv preprint arXiv:2108.13638 (2021)
17. D.I. Shuman, S.K. Narang, P. Frossard, A. Ortega, P. Vandergheynst, The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains. IEEE Signal Process. Mag. **30**(3), 83–98 (2013). https://doi.org/10.1109/MSP.2012.2235192
18. A. Sandryhaila, J.M.F. Moura, Discrete signal processing on graphs. IEEE Trans. Signal Process. **61**(7), 1644–1656 (2013). https://doi.org/10.1109/TSP.2013.2238935
19. G. Cheung, E. Magli, Y. Tanaka, M.K. Ng, Graph spectral image processing. Proc. IEEE **106**(5), 907–930 (2018). https://doi.org/10.1109/JPROC.2018.2799702
20. S. Chen, D. Tian, C. Feng, A. Vetro, J. Kovačević, Fast resampling of three-dimensional point clouds via graphs. IEEE Trans. Signal Process. **66**(3), 666–681 (2018). https://doi.org/10.1109/TSP.2017.2771730
21. C. Yang, Y. Mao, G. Cheung, V. Stankovic, K. Chan, Graph-based depth video denoising and event detection for sleep monitoring. In: 2014 IEEE 16th International Workshop on Multimedia Signal Processing (MMSP), pp. 1–6 (2014). https://doi.org/10.1109/MMSP.2014.6958802
22. A. Sandryhaila, J.M.F. Moura, Discrete signal processing on graphs: Graph fourier transform. In: 2013 IEEE International Conference on Acoustics, Speech and Signal Processing, pp. 6167–6170 (2013). https://doi.org/10.1109/ICASSP.2013.6638850
23. S. Chen, R. Varma, A. Sandryhaila, J. Kovačević, Discrete signal processing on graphs: sampling theory. IEEE Trans. Signal Process. **63**(24), 6510–6523 (2015). https://doi.org/10.1109/TSP.2015.2469645
24. D.K. Hammond, P. Vandergheynst, R. Gribonval, Wavelets on graphs via spectral graph theory. Appl. Comput. Harmon. Anal. **30**(2), 129–150 (2011). https://doi.org/10.1016/j.acha.2010.04.005
25. A.G. Marques, S. Segarra, G. Leus, A. Ribeiro, Stationary graph processes and spectral estimation. IEEE Trans. Signal Process. **65**(22), 5911–5926 (2017). https://doi.org/10.1109/TSP.2017.2739099
26. S. Zhang, Z. Ding, S. Cui, Introducing hypergraph signal processing: theoretical foundation and practical applications. IEEE Internet Things J. **7**(1), 639–660 (2020). https://doi.org/10.1109/JIOT.2019.2950213
27. S. Barbarossa, S. Sardellitti, Topological signal processing over simplicial complexes. IEEE Trans. Signal Process. **68**, 2992–3007 (2020). https://doi.org/10.1109/TSP.2020.2981920
28. D.E.O. Tzamarias, K. Chow, I. Blanes, J. Serra-Sagristà, Compression of hyperspectral scenes through integer-to-integer spectral graph transforms. Remote Sens. **11**(19), 2290 (2019)

29. N. Liu, W. Li, Q. Du, Unsupervised feature extraction for hyperspectral imagery using collaboration-competition graph. IEEE J. Select. Top. Signal Process. **12**(6), 1491–1503 (2018)
30. S. Wan, C. Gong, P. Zhong, B. Du, L. Zhang, J. Yang, Multiscale dynamic graph convolutional network for hyperspectral image classification. IEEE Trans. Geosci. Remote Sens. **58**(5), 3162–3177 (2020). https://doi.org/10.1109/TGRS.2019.2949180
31. A. Qin, C. Liu, Z. Shang, J. Tian, Spectral-spatial graph convolutional networks for semel-supervised hyperspectral image classification. In: 2018 International Conference on Wavelet Analysis and Pattern Recognition (ICWAPR), pp. 89–94 (2018). https://doi.org/10.1109/ICWAPR.2018.8521407
32. A. Qin, Z. Shang, J. Tian, Y. Wang, T. Zhang, Y.Y. Tang, Spectral-spatial graph convolutional networks for semisupervised hyperspectral image classification. IEEE Geosci. Remote Sens. Lett. **16**(2), 241–245 (2019). https://doi.org/10.1109/LGRS.2018.2869563
33. D. Hong, N. Yokoya, J. Chanussot, J. Xu, X.X. Zhu, Learning to propagate labels on graphs: an iterative multitask regression framework for semi-supervised hyperspectral dimensionality reduction. ISPRS J. Photogramm. Remote Sens. **158**, 35–49 (2019). https://doi.org/10.1016/j.isprsjprs.2019.09.008
34. D. Hong, N. Yokoya, N. Ge, J. Chanussot, X.X. Zhu, Learnable manifold alignment (lema): a semi-supervised cross-modality learning framework for land cover and land use classification. ISPRS J. Photogramm. Remote Sens. **147**, 193–205 (2019). https://doi.org/10.1016/j.isprsjprs.2018.10.006
35. D. Hong, N. Yokoya, G.-S. Xia, J. Chanussot, X.X. Zhu, X-modalnet: a semi-supervised deep cross-modal network for classification of remote sensing data. ISPRS J. Photogramm. Remote Sens. **167**, 12–23 (2020). https://doi.org/10.1016/j.isprsjprs.2020.06.014
36. D. Yao, Z. Zhi-li, Z. Xiao-feng, C. Wei, H. Fang, C. Yao-ming, W.-W. Cai, Deep hybrid: multi-graph neural network collaboration for hyperspectral image classification. Def. Technol. (2022). https://doi.org/10.1016/j.dt.2022.02.007
37. S. Jia, S. Jiang, Z. Lin, N. Li, M. Xu, S. Yu, A survey: deep learning for hyperspectral image classification with few labeled samples. Neurocomputing **448**, 179–204 (2021). https://doi.org/10.1016/j.neucom.2021.03.035
38. Y. Zhao, Y. Yuan, Q. Wang, Fast spectral clustering for unsupervised hyperspectral image classification. Remote Sens. **11**(4), 399 (2019)
39. J. Jiang, J. Ma, C. Chen, Z. Wang, Z. Cai, L. Wang, Superpca: a superpixelwise pca approach for unsupervised feature extraction of hyperspectral imagery. IEEE Trans. Geosci. Remote Sens. **56**(8), 4581–4593 (2018)
40. P. Massoudifar, A. Rangarajan, P. Gader, Superpixel estimation for hyperspectral imagery. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, pp. 287–292 (2014)
41. Y.-R. Fan, T.-Z. Huang, Hyperspectral image restoration via superpixel segmentation of smooth band. Neurocomputing **455**, 340–352 (2021)
42. Q. Yan, L. Xu, J. Shi, J. Jia, Hierarchical saliency detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2013)
43. J. Shi, J. Malik, Normalized cuts and image segmentation. IEEE Trans. Pattern Anal. Mach. Intell. **22**(8), 888–905 (2000)
44. M.-Y. Liu, O. Tuzel, S. Ramalingam, R. Chellappa, Entropy rate superpixel segmentation. In: CVPR 2011, pp. 2097–2104 (2011). IEEE
45. T.G. Kolda, B.W. Bader, Tensor decompositions and applications. SIAM Rev. **51**(3), 455–500 (2009)
46. L. De Lathauwer, B. De Moor, J. Vandewalle, A multilinear singular value decomposition. SIAM J. Matrix Anal. Appl. **21**(4), 1253–1278 (2000)
47. U. Von Luxburg, A tutorial on spectral clustering. Stat. Comput. **17**(4), 395–416 (2007)
48. R.S. Boyer, J.S. Moore, Mjrty–a fast majority vote algorithm, pp. 105–117. Springer (1991)
49. J. Weston, C. Watkins, *Multi-class support vector machines* (Technical report, Citeseer, 1998)
50. M. De Domenico, V. Nicosia, A. Arenas, V. Latora, Structural reducibility of multilayer networks. Nat. Commun. **6**(1), 1–9 (2015)
51. X. He, D. Cai, S. Yan, H.-J. Zhang, Neighborhood preserving embedding. In: Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1, vol. 2, pp. 1208–1213 (2005). IEEE
52. Y. Zhou, J. Peng, C.L.P. Chen, Dimension reduction using spatial and spectral regularized local discriminant embedding for hyperspectral image classification. IEEE Trans. Geosci. Remote Sens. **53**(2), 1082–1095 (2015)
53. S. Prasad, L.M. Bruce, Limitations of principal components analysis for hyperspectral target recognition. IEEE Geosci. Remote Sens. Lett. **5**(4), 625–629 (2008)
54. W. Li, S. Prasad, J.E. Fowler, L.M. Bruce, Locality-preserving dimensionality reduction and classification for hyperspectral image analysis. IEEE Trans. Geosci. Remote Sens. **50**(4), 1185–1198 (2012)
55. X. Dong, D. Thanou, M. Rabbat, P. Frossard, Learning graphs from data: a signal representation perspective. IEEE Signal Process. Mag. **36**(3), 44–63 (2019). https://doi.org/10.1109/MSP.2018.2887284