A Fast and Accurate Similarity-Constrained Subspace Clustering Algorithm for Hyperspectral Image

Carlos Hinojosa, *Student Member, IEEE*, Esteban Vera, *Member, IEEE*, and Henry Arguello, *Senior Member, IEEE*

***Abstract*—Accurate unsupervised classification of hyperspectral images (HSIs) is challenging and has drawn widespread attention in remote sensing due to its inherent complexity. Although significant efforts have been made to develop a variety of methods, most of them rely on supervised strategies. Subspace clustering methods, such as sparse subspace clustering (SSC), have become a popular tool for unsupervised learning due to their high performance. However, the computational complexity of SSC methods prevents their use on full HSIs. Furthermore, since SSC ignores the spatial information in the HSIs, its discrimination capability is limited, hampering the clustering results' spatial homogeneity. To address these two relevant issues, in this article, we propose a fast algorithm that obtains a sparse representation coefficient matrix by first selecting a small set of pixels that best represent their neighborhood. Then, it performs spatial filtering to enforce the connectivity of neighboring pixels and uses fast spectral clustering to get the final clustering map. Extensive simulations with our proposed method demonstrate its effectiveness in unsupervised HSI classification, obtaining remarkable high clustering performance compared with state-of-the-art SSC-based algorithms and even novel unsupervised-deep-learning-based methods. Besides, the proposed method is up to three orders of magnitude faster than SSC when clustering more than 2×10^4 spectral pixels.**

***Index Terms*—Hyperspectral image clustering, spectral–spatial classification, subspace clustering, unsupervised learning.**

I. Introduction

SPECTRAL remote sensing systems acquire the Earth's surface information by sensing a large amount of spatial data at different electromagnetic radiation frequencies. Hyperspectral images (HSIs) are commonly regarded as 3-D datasets or data cubes with two dimensions in the spatial domain (x, y) and one in the spectral domain (λ) [1]. As depicted in Fig. 1, every spatial location in an HSI is represented by a vector whose values correspond to the intensity at different spectral bands. These vectors are also known as the spectral signature of the pixels or spectral pixels. Since different materials usually reflect electromagnetic energy differently at specific wavelengths [1], the information provided by the spectral signatures allows distinguishing different physical materials and objects within an image. The research in hyperspectral remote sensing is mainly focused on developing new and automatic methods for hyperspectral imagery analyses, including spectral classification [2], image denoising [3], and spectral clustering. In particular, the efficient clustering or unsupervised classification of HSIs is an important task for many practical applications, such as precision agriculture [4], vegetation classification [5], urban land use mapping [6], change detection [7], monitoring and management of the environment [8], [9], and security and defense issues [10].

Accurate HSI clustering is challenging due to the high-dimensional feature space, and it has drawn widespread attention in remote sensing [11], [12]. In the past decade, significant efforts have been made in the development of numerous HSI classification methods; however, most of them rely on supervised approaches [13], [14]. More recently, with the blooming of deep learning techniques for extensive data analysis, several deep neural networks have been developed to extract high-level features of HSIs achieving state-of-the-art supervised classification performance [15]. However, the success of such deep learning approaches hinges on a large amount of labeled data, which is not always available and often prohibitively expensive to acquire. As a result, the remote sensing community is currently focused on developing unsupervised methods that can adapt to new conditions without requiring a massive amount of data [16].

The sparse subspace clustering (SSC) algorithm is one of the most successful unsupervised learning methods in the literature. SSC exploits the fact that high-dimensional data can be well represented as the union of low-dimensional subspaces. Under this assumption, SSC captures the relationship among all data points by exploiting the *self-expressiveness* property [17]. This property states that each data point in a union of subspaces can be written as a linear combination of other points from its own subspace. Then, the set of solutions is restricted to be sparse by minimizing the ℓ_1 norm. Finally, an affinity matrix is built using the obtained sparse coefficients, and the normalized spectral clustering algorithm [18] is applied to achieve the final clustering map.

Assuming that spectral pixels with a similar spectrum approximately belong to the same low-dimensional structure, the SSC algorithm has been successfully applied to HSIs [19]–[25]. Despite the great success of SSC in HSI clustering, two main problems are well known.

Manuscript received July 9, 2021; revised September 4, 2021; accepted October 6, 2021. Date of publication October 14, 2021; date of current version November 2, 2021. This work was supported by the Fondo Nacional de Desarrollo Científico y Tecnológico under Grant 1181943. *(Corresponding author: Carlos Hinojosa.)*

Carlos Hinojosa and Henry Arguello are with the Department of System Engineering and Informatics, Universidad Industrial de Santander, Bucaramanga 680002, Colombia (e-mail: carlos.hinojosa@saber.uis.edu.co; henarfu@uis.edu.co).

Esteban Vera is with the School of Electrical Engineering, Pontificia Universidad Católica de Valparaíso, Valparaíso 2340000, Chile (e-mail: esteban.vera@pucv.cl).

Digital Object Identifier 10.1109/JSTARS.2021.3120071

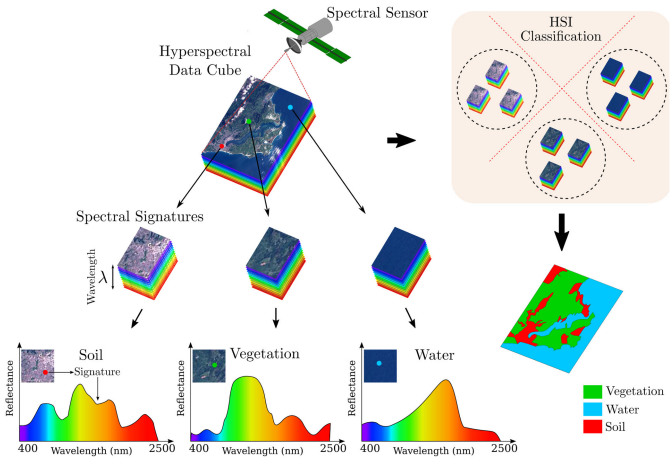


Fig. 1. HSI classification. Since different materials usually reflect electromagnetic energy differently at specific wavelengths, the spectral signatures allow discriminating materials within an HSI.

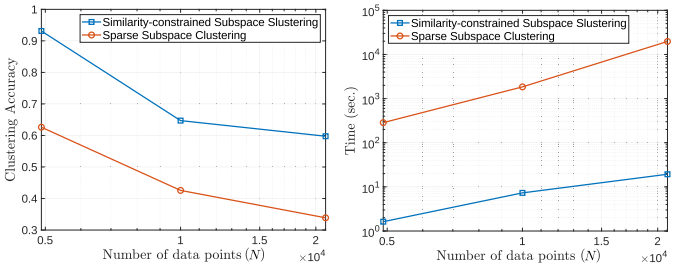


Fig. 2. Clustering accuracy (left) and running time (right) of the SSC algorithm compared with our proposed method for HSI clustering. In this example, we vary the number of data points (N) by performing the two subspace clustering algorithms on the full image and two ROIs of the Indian Pines dataset (see Section IV). The first ROI has $N = 4900$ pixels and $k = 4$ classes; the second has $N = 10000$ pixels and $k = 12$ classes; and the whole Indian Pines image has $N = 21025$ pixels and $k = 17$ classes.

1) The overall computational complexity of SSC prohibits its usage on large HSI datasets. For instance, given an HSI with N_r rows, N_c columns, and L spectral bands, SSC needs to compute the $N \times N$ sparse coefficient matrix corresponding to $N = N_r N_c$ spectral pixels, whose computational complexity is $O(LN^3)$. Moreover, after building the affinity matrix, spectral clustering performs an eigenvalue decomposition over the $N \times N$ graph Laplacian matrix, which also has cubic time complexity, or quadratic using approximation algorithms [26] (see Fig. 2, right).
2) Under the context of HSI, the SSC model only captures the relationship of pixels by analyzing the spectral features without considering the spatial information. Indeed, the sparse coefficient matrix obtained by SSC should be piecewise smooth since spectral pixels belonging to the same land cover material are arranged in a common region; hence, there is a spatial relationship between the representation coefficient vector of one pixel and its neighbors.

Article Contribution: In this article, we are interested in the unsupervised classification of HSI using SSC. Since SSC methods require to express each data point as a linear combination of all other points in the dataset, our main observation is that we can take advantage of the spatial–spectral properties of HSI to select the most representative spectral pixels over small regions and then constrain the remaining data points to be represented as a linear combination of them. This significantly reduces the clustering time and increases the accuracy. Specifically, we propose an efficient algorithm for selecting the most representative pixels of each subset by minimizing the maximum representation cost of the data. After this procedure, we concatenate the obtained most representative spectral pixels to form a matrix. We solve an optimization problem using this matrix to get the coefficients that encodes information about similarities between each subset's most representative spectral pixels and the whole HSI. Finally, we propose to enhance the obtained coefficient matrix via 2-D smoothing convolution before applying a fast spectral clustering algorithm that provides the final clustering map in a significantly less amount of time. The motivation of using such 2-D convolution is to enforce the piecewise smoothness in the coefficient matrix. In essence, the proposed method enforces the connectivity in the affinity matrix and then efficiently obtains spectral embedding without the need to compute the eigenvalue decomposition, which has a computational complexity of $O(N^3)$.

Increasing the number of data points and the classes enlarges the computation time and makes clustering more challenging. The proposed method, shown with the blue line in Fig. 2, can be up to three orders of magnitude faster than SSC and outperforms it in terms of accuracy when clustering more than 2×10^4 spectral pixels. This article evaluates and compares our approach on three real remote sensing HSIs with different imaging environments and spectral–spatial resolution. Throughout this article, we use the words "spectral pixels" and "spectral signatures" interchangeably.

II. Related Works

In this section, we review some related works from two points of view: *efficient general-purpose SSC-based methods* and *SSC-based methods for HSI clustering*. Considering a given collection of N data points $\mathbf{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$ that lie in the union of k linear subspaces of $\mathbb{R}^D$, SSC expresses each data point $\mathbf{x}_j$ as a linear combination of all other points in $\mathbf{X}$, i.e., $\mathbf{x}_j = \sum_{i \neq j} c_{ij} \mathbf{x}_i$, where c_{ij} is nonzero only if $\mathbf{x}_i$ and $\mathbf{x}_j$ are from the same subspace, for $(i, j) \in \{1, \ldots, N\}$. Such representations $\{c_{ij}\}$ are called *subspace preserving*. Once obtained the $\{c_{ij}\}$ representations, an affinity matrix between any pair of points $\mathbf{x}_i$ and $\mathbf{x}_j$ is defined as $A_{ij} = |c_{ij}| + |c_{ji}|$, and it is used as input for the spectral clustering algorithm to infer the clustering of the data [17], [18]. Although the representation produced by SSC is guaranteed to be subspace preserving, the affinity matrix may lack *connectedness* [27], i.e., the data points from the same subspace may not form a connected component of the affinity graph due to the sparseness of the connections.

A. Efficient General-Purpose SSC-Based Methods

Considering the self-expressiveness property, an early approach to address the SSC scalability issue assumes that a small number of data points can represent the whole dataset without loss of information. Then, Peng *et al.* [28] proposed

the *scalable sparse subspace clustering* (SSSC) algorithm to cluster a small subset of the original data and then classify the rest of the data based on the learned groups. However, this strategy is suboptimal since it sacrifices clustering accuracy for computational efficiency.

In [29], You *et al.* replaced the ℓ_1 optimization in the original SSC algorithm [17] with greedy pursuit, e.g., orthogonal matching pursuit (OMP) [30], for sparse self-representation [31]. While SSC-OMP improves the time efficiency of SSC by several orders of magnitude, it significantly loses clustering accuracy [32]. Besides, SSC-OMP also suffers from the connectivity issue presented in the original SSC algorithm. To solve this issue, You *et al.* [33] proposed to mixture the ℓ_1 and ℓ_2 norms to take advantage of subspace preserving of the ℓ_1 norm and the dense connectivity of the ℓ_2 norm. Specifically, the algorithm named ORacle Guided Elastic Net solver (ORGEN) is proposed to identify a support set for each sample. However, in this approach, a convex optimization problem is solved several times for each sample, limiting the algorithm's scalability.

More recent works use a different subset selection method for subspace clustering. In particular, the method named scalable and robust SSC (SR-SSC) [34] selects a few sets of anchor points using a randomized hierarchical clustering method. Then, within each set of anchor points, it solves the LASSO [35] problem for each data point, allowing only anchor points to have nonzero weights.

Similar to the SSC-OMP paper, You *et al.* [36] proposed an approximation algorithm [37] to solve the SSC optimization problem. Specifically, instead of using all the dataset $\mathbf{X}$, the exemplar-based subspace clustering (ESC-FFS) algorithm in [36] selects a small subset $\tilde{\mathbf{X}} \subseteq \mathbf{X}$ that represents all data points, and then, each point is expressed as a linear combination of points in $\tilde{\mathbf{X}} \in \mathbb{R}^{D\times M}$, where $M < N$. In particular, the selection of $\tilde{\mathbf{X}}$ is obtained by using the farthest first search (FFS) algorithm, which is a modified version of the farthest-first traversal algorithm, where the main difference is the used distance metric [37]. The authors propose to construct $\tilde{\mathbf{X}}$ by first performing random sampling to select a base point and then progressively add new representative data points using the defined metric.

In general, the previously described algorithms provide an acceptable subspace clustering performance on large datasets. However, these general-purpose methods do not fully exploit the complex structure of remotely sensed HSIs, ignoring their rich spatial information, which could boost the accuracy of these algorithms.

B. SSC-Based Methods for HSI Clustering

Some SSC-based methods have been proposed for unsupervised HSI classification, which take advantage of the neighboring spatial information but still present the efficiency issue of SSC. Under the context of HSIs, the $N_r \times N_c \times L$ 3-D image data cube can be rearranged into a 2-D matrix $\mathbf{X} \in \mathbb{R}^{D\times N}$ to apply the SSC algorithm, where $N = N_r N_c$ and $D < L$ is the number of features extracted from the spectral signatures after using principal component analysis (PCA) [13]. Taking into account that the spectral pixels belonging to the same land cover material are arranged in common regions, different works [19], [21]–[24], [38], [39] aim at obtaining a piecewise smooth sparse coefficient matrix to incorporate such contextual dependence. In particular, S-SSC [23] helps to guarantee spatial smoothness and reduce the representation bias by adding a regularization term in the SSC optimization problem, which enforces a local averaging constraint on the sparse coefficient matrix. More recently, Hinojosa *et al.* [19] have proposed the 3DS-SSC algorithm, which incorporates a 3-D Gaussian filter in the optimization problem to perform a 3-D convolution on the sparse coefficients, obtaining a piecewise-smooth representation matrix. Also, another recent work proposes a graph convolutional subspace clustering (GCSC) framework [20] that efficiently processes graph data by modeling information from neighbor samples (or nodes). Then, the graph is used as a dictionary for the subsequent affinity learning and finally outputting classification results by a clustering model. In general, these methods are slow since they load and process all the data points, hence having large memory dependence. Therefore, these methods commonly present clustering results on small regions of interest (ROIs) from the full HSI in their original manuscripts. Aiming to be fair, we compare our proposed algorithm with these methods using the same commonly used ROIs in the experimental section.

III. Fast and Accurate Similarity-Constrained Subspace Clustering (SC-SSC)

This section presents our proposed subspace clustering algorithm for unsupervised HSI classification that incorporates both properties: it is efficient and takes advantage of the spatial information of HSIs to boost the clustering accuracy. In general, we exploit the self-representation property within subsets of neighboring similar pixels to select the most representative data points of the whole HSI. Then, we enhance the sparse representation and perform fast spectral clustering to obtain the segmentation result. We show the complete workflow of the proposed method in Fig. 3.

A. Similarity-Constrained Spectral Pixel Selection

As neighboring spatial pixels commonly belong to the same land cover material, our proposed method aims to select a small subset of pixels that best represent their neighborhood. In this regard, we first propose to group all the spectral pixels with similar spectrum into small spatial regions. In general, this preprocessing step can be done in different ways. In this article, we particularly adopted the simple linear iterative clustering (SLIC) [40] algorithm to generate the guidance region map due to its efficiency. In particular, we perform PCA to retrieve the three principal components of $\mathbf{X}$ and form the matrix $\mathbf{X}_{\text{PCA}} \in \mathbb{R}^{3\times N}$. Then, we use the SLIC algorithm [40] to obtain a regions map $\tilde{\mathbf{m}} \in \mathbb{R}^N$ from $\mathbf{X}_{\text{PCA}}$ such that $\tilde{m}_j \in \{1, \ldots, E\}$, where E is the number of segments. For instance, if $\tilde{m}_j = e$, it means that the spectral pixel $\mathbf{x}_j$ belongs to the segment e. Note

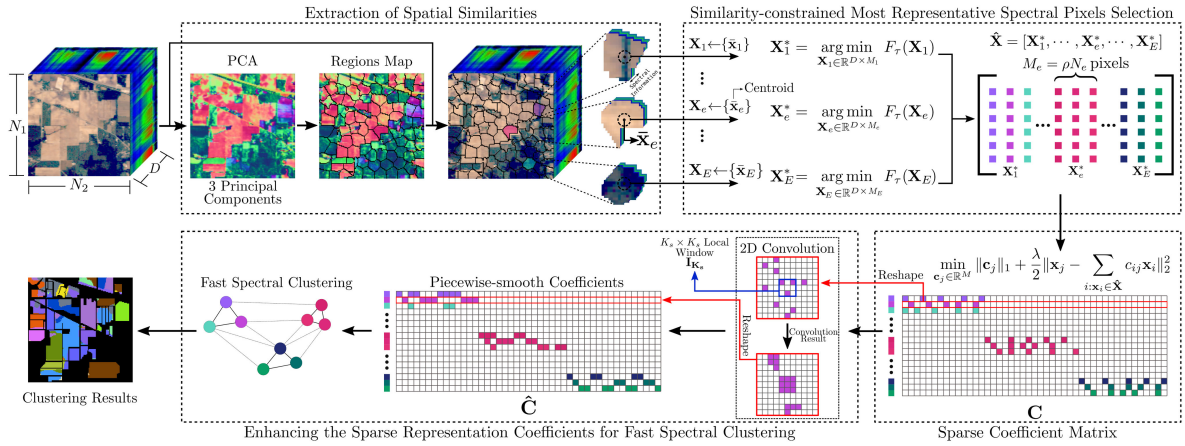


Fig. 3. Our proposed workflow is composed of four stages. In the first stage, we group spectral pixels with similar values into regions. For this step, we first apply PCA to obtain the three principal components of the HSI and adopt the SLIC algorithm to generate a guidance map and use it to group similar spectral pixels into regions. In the second stage, we use our proposed Algorithm 1 to select the most representative spectral pixels from each region or subset and then stacked them as columns in the matrix $\hat{\mathbf{X}}$. In the third stage, we solve (4) to obtain the $\{\mathbf{c}_j\}$ vectors. We propose to further improve the connectivity of matrix $\mathbf{C} = [\mathbf{c}_1, \ldots, \mathbf{c}_N]$ by performing a 2-D convolution with a $K_s \times K_s$ kernel, hence obtaining a piecewise-smooth coefficient matrix. We obtain the final data segmentation via fast spectral clustering. The computational complexity of the overall algorithm is $O(\rho^2 N^3)$ (see Section III-C2).

that PCA is only performed to obtain $\tilde{\mathbf{m}}$ from $\mathbf{X}_{\text{PCA}}$ via SLIC; then, we use $\tilde{\mathbf{m}}$ to select the most representative spectral pixels $\mathbf{x}_j$ from $\mathbf{X}$ within each segment.

Let $\mathbf{p}_e \in \mathbb{R}^{N_e}$ be the vector containing the indices of the N_e most similar spectral pixels belonging to the subset e. We are interested in selecting the $M_e = \lfloor \rho N_e \rfloor$ most representative pixels from each subset, where $\rho \in (0, 1)$. Taking advantage of the self-expressiveness property, the selection of the pixels within each neighborhood e is obtained by searching for a subset $\mathbf{X}_e^* \subseteq \mathbf{X}$ that minimizes

$$\mathbf{X}_e^* = \underset{\mathbf{X}_e \in \mathbb{R}^{D \times M}}{\arg\min}\, F_\tau(\mathbf{X}_e) \tag{1}$$

where F_τ is the *self-representation* cost function defined as

$$F_\tau(\mathbf{X}_e) := \sup_{\mathbf{x}_j \in \mathbf{X} \,:\, j \in \mathbf{p}_e} f_\tau(\mathbf{x}_j, \mathbf{X}_e). \tag{2}$$

The metric function $f_\tau(\mathbf{x}_j, \mathbf{X}_e)$ geometrically measures how well a data point $\mathbf{x}_j \in \mathbf{X} : j \in \mathbf{p}_e$ can be represented by the subset $\mathbf{X}_e$, and we define it as

$$f_\tau(\mathbf{x}_j, \mathbf{X}_e) := \min_{\mathbf{c}_j \in \mathbb{R}^N} \|\mathbf{c}_j\|_1 + \frac{\tau}{2} \left\| \mathbf{x}_j - \sum_{i:\mathbf{x}_i \in \mathbf{X}_e} c_{ij} \mathbf{x}_i \right\|_2^2 \tag{3}$$

where $\tau \in (1, \infty)$ is a parameter. Note that with (2), we constrain (1) to search only for pixels $\mathbf{x}_j$ within the subset e, using the vector $\mathbf{p}_e$. To efficiently solve (1) for each subset e, we propose the algorithm described in Algorithm 1. Note that, instead of using a random initialization, we select the centroid spectral pixel $\bar{\mathbf{x}}_e$ as the initialization since it is the most similar point, in the Euclidean distance, to all other data points in e. The search space constraint, given by dividing the HSI into subsets and selecting the centroid spectral pixel, speeds up the acquisition of the representative spectral pixels.

Algorithm 1: Similarity-Constrained Spectral Pixels Selection.

Input : Data $\mathbf{X} \in \mathbb{R}^{D \times N}$, Indices vector $\mathbf{p}_e \in \mathbb{R}^{N_e}$, Parameters $0 < \rho < 1$, and $\tau > 1$.
Output: $\mathbf{X}_e \in \mathbb{R}^{D \times \lfloor \rho N_e \rfloor}$.
Function *Data_Selection*$(\mathbf{X}, \mathbf{p}_e, \rho, \tau)$

$\bar{\mathbf{x}}_e \leftarrow \text{centroid}(\{\mathbf{x}_j \in \mathbf{X} : j \in \mathbf{p}_e\})$
$\mathbf{X}_e^{(1)} \leftarrow \{\bar{\mathbf{x}}_e\}$
▷ $(\mathbf{p}_e)_k$ gets the k element of the vector $\mathbf{p}_e$.
Compute $b_k = f_\tau(\mathbf{x}_j, \mathbf{X}_e^{(1)})$ for $k = 1, \cdots, N_e$, and $j = (\mathbf{p}_e)_k$.
$M_e \leftarrow \lfloor \rho N_e \rfloor$
for $i = 1, \cdots, M_e - 1$ **do**
  Let $o_1, \cdots, o_{N_e}$ be an ordering of $1, \cdots, N_e$ such that $b_{o_p} \geq b_{o_q}$ when $p < q$.
  Initialize $max\_cost = 0$.
  for $k = 1, \cdots, N_e$ **do**
    Set $b_{o_k} = f_\tau(\mathbf{x}_{o_k}, \mathbf{X}_e^{(i)})$.
    if $b_{o_k} > max\_cost$ **then**
      Set $max\_cost = b_{o_k}$, and $new\_index = o_k$.
    if $k = N_e$ *or* $max\_cost \geq b_{o_{k+1}}$ **then**
      break
  $\mathbf{X}_e^{(i+1)} = \mathbf{X}_e^{(i)} \cup \{\mathbf{x}_{new\_index}\}$
return $\mathbf{X}_e$

B. Enhancing the Sparse Representation Coefficients for Fast Spectral Clustering

Once obtained the most representative spectral pixels from each subset, we build the matrix $\hat{\mathbf{X}}$ by stacking the results as columns, i.e., $\hat{\mathbf{X}} = [\mathbf{X}_1, \ldots, \mathbf{X}_E]$. Then, the sparse coefficient matrix $\mathbf{C}$ of size $M \times N$, with $M = \lfloor \rho N \rfloor$, can be obtained by solving the following optimization problem:

$$\min_{\mathbf{c}_j \in \mathbb{R}^M} \|\mathbf{c}_j\|_1 + \frac{\tau}{2} \|\mathbf{x}_j - \sum_{i:\mathbf{x}_i \in \hat{\mathbf{X}}} c_{ij} \mathbf{x}_i\|_2^2 \quad \forall\, \mathbf{x}_j \in \mathbf{X}. \tag{4}$$

Algorithm 2: Our Proposed Algorithm For HSI Clustering (SC-SSC).

Input : The hyperspectral image in matrix form $\mathbf{X} \in \mathbb{R}^{D\times N}$, parameters $\tau > 1$, $0 < \rho < 1$, $E > 1$, $K_s > 1$.

Output: The segmentation of $\mathbf{X}$.

$\mathbf{X}_{PCA} \in \mathbb{R}^{3\times N} \leftarrow PCA(\mathbf{X})$
$\tilde{\mathbf{M}} \leftarrow Extract\_Regions\_Map(\mathbf{X}_{PCA})$
$\tilde{\mathbf{m}} \leftarrow \text{vec}(\tilde{\mathbf{M}})$
$\hat{\mathbf{X}}^{(e)} \leftarrow \emptyset$
for $e \leftarrow 1$ **to** $E-1$ **do**
  $\mathbf{p}_e = \{j : \tilde{\mathbf{m}}_j = e, \forall j \in \{1, \cdots, N\}\}$
  $\mathbf{X}_e \leftarrow Data\_Selection(\mathbf{X}, \mathbf{p}_e, \rho, \tau)$
  $\hat{\mathbf{X}}^{(e+1)} = \hat{\mathbf{X}}^{(e)} \cup \mathbf{X}_e$
Compute $\mathbf{C} = [\mathbf{c}_1, \cdots, \mathbf{c}_N]$ by solving Eq. (4).
$\mathbf{I}_{K_s} = (1/K_s^2) \cdot \mathbf{1}$ ▷ 1: $K_s \times K_s$ all-ones matrix.
$\hat{\mathbf{C}} = \mathcal{G}(\mathbf{C}, \mathbf{I}_{K_s})$ ▷ 2D Convolution Operation
$\tilde{\mathbf{C}} = [\hat{\mathbf{c}}_1/\|\hat{\mathbf{c}}_1\|_2, \cdots, \hat{\mathbf{c}}_N/\|\hat{\mathbf{c}}_N\|_2]$
$\alpha = \sum_{j=1}^{N} \tilde{\mathbf{c}}_j$
$\mathbf{D} = \text{diag}(\tilde{\mathbf{C}}^T\alpha)$
Run k-means clustering algorithm on the top k right singular vectors of $\tilde{\mathbf{C}}\mathbf{D}^{-1/2}$ to obtain the segmentation of $\mathbf{X}$.
return *The cluster assignments of* $\mathbf{X}$

Note that $\mathbf{C}$ encodes information about the similarities between $\hat{\mathbf{X}}$ and $\mathbf{X}$. Besides, each row of $\mathbf{C}$ contains the representation coefficient distribution of the whole image with respect to a single representative pixel. Consider that the spectral pixels belonging to the same land cover material should be regionally distributed in the image, i.e., two spatially neighboring pixels in an HSI usually have a high probability of belonging to the same class. Then, according to the self-expressiveness property, their representation coefficients should also be very close concerning the same basis; hence, each row of $\mathbf{C}$ should be piecewise smooth. Therefore, to further improve the structure of matrix $\mathbf{C}$, we propose to apply a 2-D smoothing convolution on its coefficients. This is a postprocessing procedure, and it is performed before spectral clustering. Given a blur kernel matrix $\mathbf{I}_{K_s}$ of size $K_s \times K_s$, we will denote the 2-D convolution process as $\hat{\mathbf{C}} = \mathcal{G}(\mathbf{C}, \mathbf{I}_{K_s})$. Specifically, as depicted in Fig. 3 within the dashed blue line, we propose to perform $\mathcal{G}$ by first reshaping each row of $\mathbf{C}$ to a window of size $N_r \times N_c$, which corresponds to the spatial dimensions of the HSI, and then conducting the convolution with $\mathbf{I}_{K_s}$. Finally, the convolution result is rearranged back as a row vector of the piecewise-smooth coefficient matrix $\hat{\mathbf{C}} = [\hat{\mathbf{c}}_1, \ldots, \hat{\mathbf{c}}_N] \in \mathbb{R}^{M\times N}$.

Finally, we use fast spectral clustering to efficiently obtain the spectral embedding of the input data. Specifically, let us consider the columns of $\tilde{\mathbf{C}} = [\tilde{\mathbf{c}}_1, \ldots, \tilde{\mathbf{c}}_N] \in \mathbb{R}^{M\times N}$, where $\tilde{\mathbf{c}}_j = |\hat{\mathbf{c}}_j|/\|\hat{\mathbf{c}}_j\|_2$, and compute the ith element of $\mathbf{D}$ as

$$(\mathbf{D})_i = \sum_{j=1}^{N} A_{ij} = \sum_{j=1}^{N} \tilde{\mathbf{c}}_i^T \tilde{\mathbf{c}}_j = \tilde{\mathbf{c}}_i^T \sum_{j=1}^{N} \tilde{\mathbf{c}}_j = \text{diag}(\tilde{\mathbf{C}}^T\boldsymbol{\alpha})_i \quad (5)$$

where $\boldsymbol{\alpha} = \sum_{j=1}^{N} \tilde{\mathbf{c}}_j \in \mathbb{R}^M$. Next, we can find the eigenvalue decomposition of $\mathbf{D}^{-1/2}\mathbf{A}\mathbf{D}^{-1/2}$ by computing the singular value decomposition (SVD) [41] of $\tilde{\mathbf{C}}\mathbf{D}^{-1/2} \in \mathbb{R}^{M\times N}$. Finally, the segmentation of the data can be obtained by running the k-means algorithm on the top k right singular vectors for $\tilde{\mathbf{C}}\mathbf{D}^{-1/2} = \mathbf{U}\boldsymbol{\Sigma}\mathbf{P}^T$. As a result, the computational complexity of spectral clustering in our framework is linear with respect to the size of the data N. Our proposed SC-SSC method is summarized in Algorithm 2.

C. Analysis of the Proposed Method

1) Subspace-Preserving Property and Connectivity: As mentioned in Section II, one of the main requirements for the success of subspace clustering methods is that the optimization process recovers a subspace-preserving solution. Specifically, the nonzero entries of the sparse representation vector $\mathbf{c}_j$ should be related only to the intrasubspace samples of $\mathbf{x}_j$. Indeed, as the following definition states, the representation coefficients among intrasubspace data points are always larger than those among intercluster points.

Definition 1 (Intrasubspace projection dominance, IPD [42]): The IPD property of a coefficient matrix $\mathbf{C}$ indicates that for all $\mathbf{x}_u, \mathbf{x}_v \in \mathcal{S}$ and $\mathbf{x}_q \notin \mathcal{S}$, where $u, v, q \in \{1, \ldots, N\}$, and $\mathcal{S}$ is a subspace of $\mathbf{X}$, we have $C_{uv} \geq C_{uq}$.

Since the proposed method selects the most representative spectral pixels for each subset e based on the self-representation property, it is expected that each subset is subspace preserving, i.e., c_{ij} is nonzero only if $\mathbf{x}_i$ and $\mathbf{x}_j$, for $i, j \in \mathbf{p}_e$, belong to the same subspace $\mathcal{S}$. Furthermore, note that it is very probable that a subset e has more spectral pixels from the same class due to the spatial dependence in the HSI; then, the resulting coefficient vector will have large values for those spectral pixels within e. Therefore, the strategy adopted in the proposed method will improve the structure of the vectors $\mathbf{c}_j$ obtained by (4) and will improve the probability that $\mathbf{c}_j$ satisfies the IPD.

Besides, using the 2-D smoothing convolution procedure $\mathcal{G}(\mathbf{C}, \mathbf{I}_{K_s})$, the proposed method improves the connectivity of the data points by preserving the most significant values in the coefficient matrix $\mathbf{C}$ and reducing the small or noisy isolated values, based on the IPD property [42]. Then, the resulting matrix $\hat{\mathbf{C}}$ will have localized neighborhoods in the sparse codes making the representation coefficients of spatially neighboring pixels very close as well, following our main assumption in Section III-B.

2) Computational Complexity Analysis: As shown in Fig. 3, the proposed method mainly involves four stages: the extraction of spatial similarities, the selection of similarity-constrained representative spectral pixels, the sparse coefficient matrix estimation by solving (4), and enhancing the representation coefficients for fast spectral clustering. Given an HSI in matrix form $\mathbf{X} \in \mathbb{R}^{D\times N}$ and E subsets $\mathbf{X}_e \subseteq \mathbf{X}$ of dimensions $D \times M_e$, with $M_e = \rho N_e$, we will show the complexity of each stage before establishing the total complexity of Algorithm 2. Specifically, in the first stage, we acquire the segmentation map $\tilde{\mathbf{m}}$ for an HSI. Such a procedure involves computing PCA over

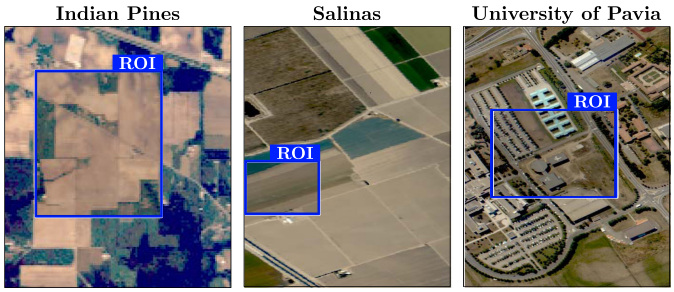


Fig. 4. False-color images and ROIs for the three real remote sensing images used in our experiments.

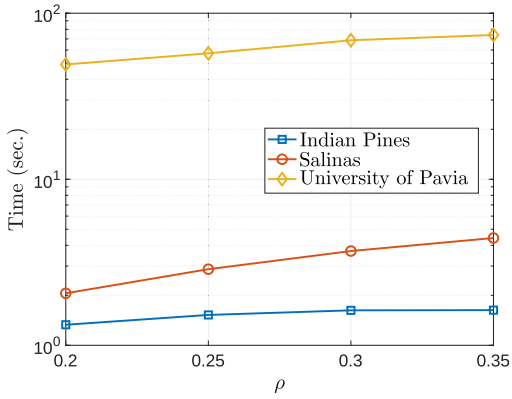


Fig. 5. Running time (in seconds) as a function of the ρ parameter.

$\mathbf{X}$ to retrieve only the three principal components, which takes $O(N)$, and performing SLIC superpixels [40], which also has linear time complexity $O(N)$. The second stage requires to execute Algorithm 1, which has $O(\rho N_e^2)$ time complexity over E subsets; then, the overall complexity of this stage will be $O(\rho \max(N_1^2, \ldots, N_E^2))$. The third stage entails solving (4), which is a LASSO problem that can be efficiently computed in $O(M^2\, N)$ using the LARS algorithm [43]. Finally, in the last stage, the 2-D convolution takes $O(N)$ as $K_s \ll N$ and, since for the spectral clustering we only need the k largest singular values, we can use the truncated SVD, which takes $O(k^2\, N)$. Thus, the overall complexity of this stage is $O(k^2\, N)$. Therefore, the complexity of Algorithm 2 will be dominated by the complexity of the third stage; hence, it will run in $O(M^2\, N) = O(\rho^2\, N^3)$, where $\rho \in (0, 1)$.

IV. Experimental Evaluation

In this section, we show the performance of SC-SSC[1] for unsupervised HSI classification. The sparse optimization problem in (4) is solved by the LASSO version of the LARS algorithm [43] implemented in the SPAMS package [44]. All the experiments were run on an Intel Core i7 9750H CPU (2.60 GHz, six cores), with 32 GB of RAM.

A. Setup

1) Datasets: The proposed subspace clustering approach (SC-SSC) was tested on three well-known HSIs[2] with different imaging environments (see Fig. 4). The *Indian Pines* hyperspectral dataset has 145 × 145 pixels and 200 spectral bands in the range of 0.4–2.5 μm and includes 16 land-cover classes. The second scene, *Salinas*, has 512 × 217 pixels and 204 spectral bands in the range of 0.24–2.40 μm and includes 16 land-cover classes. The third scene, *University of Pavia*, comprises 610 × 340 pixels and has 103 spectral bands with spectral coverage ranging from 0.43 to 0.84 μm and includes nine classes.

As mentioned in Section II, most SSC-based methods for HSI are inefficient and have a large memory dependence; hence, they suffer from out-of-memory problems during their execution. In order to make a fair comparison with such nonscalable methods, we separately take a frequently used ROI of these datasets for evaluation, as done in several previous works [19], [20], [23]. We show the selected ROIs in Fig. 4. The Indian Pines ROI has a size of 70 × 70 pixels, which includes four main land-cover classes: corn-no-till, grass, soybeans-no-till, and soybeans-min-till. The Salinas ROI comprises 83 × 83 pixels and includes six classes: broccoli-1, corn-senesced, lettuce-4wk, lettuce-5wk, lettuce-6wk, and lettuce-7wk. Finally, the University of Pavia ROI is composed of 200 × 200 pixels and includes all the classes (nine) as in the full image: asphalt, meadows, gravel, trees, metal sheets, bare soil, bitumen, bricks, and shadows. For all experiments, we reduce the spectral dimensions of each image using PCA to $D = 0.25L$, where L is the number of spectral bands. Then, we rearrange the data cube to form a matrix $\mathbf{X} \in \mathbb{R}^{D \times N}$ and normalize the columns to have unit ℓ_2 norm.

2) Baselines and Evaluation Metrics: We compare our approach with several methods and separate them into four categories: *SSSC-based methods*, *nonscalable SSC-based methods*, *unsupervised-deep-learning-based methods*, and *traditional clustering methods*. Specifically, under the *SSSC-based methods*, we compare our approach with the algorithms: SSC-OMP [29], SSSC [28], ESC-FFS [36], and SR-SSC [34]. Under the *nonscalable SSC-based methods*, we compare our approach with the algorithms: S-SSC [23], ORGEN [33], 3DS-SSC [19], and EGCSC [20]. We also show the results with SSC as an additional reference. For the sake of completeness, we also compare our approach with VAE [45], AE-GRU [45], AE-LSTM [45], and the 3D-CAE [46], which fall under the *unsupervised-deep-learning-based methods*, and the K-means [47], fuzzy c-means (FCM) [48], FCM_S1 [49], and FCM_S2 [49], which are *traditional clustering methods*.

To compare the clustering performance of our model, we rely on five standard metrics: user's accuracy (UA), average accuracy (AA), overall accuracy (OA), Kappa coefficient, and normalized mutual information (NMI) [50], [51]. In particular, UA, AA, OA, and Kappa coefficient can be obtained employing an error matrix (a.k.a confusion matrix) [50]. UA represents the clustering accuracy of each class, while AA is the mean of UA, and OA is computed by dividing the number of correctly

[1]Code available at: https://link.carloshinojosa.me/SC-SSC.

[2][Online]. Available: https://link.carloshinojosa.me/hsi_scenes

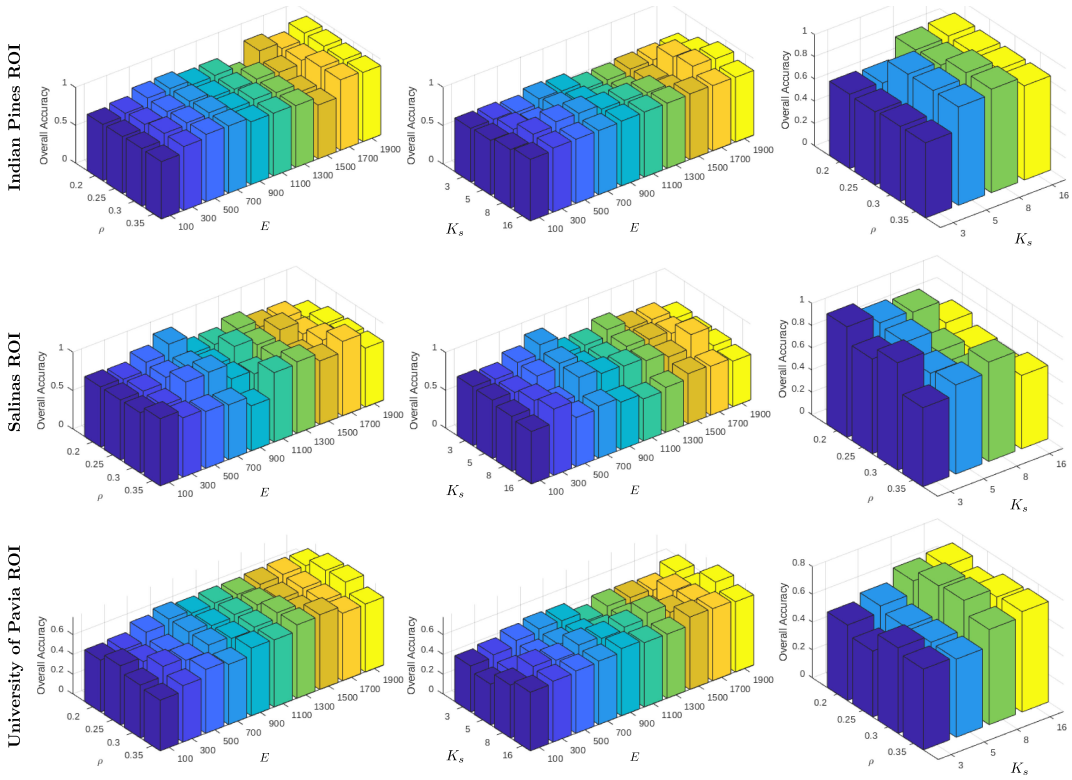


Fig. 6. Analysis of influence of parameters ρ, E, and K_s in Algorithm 2. Each row presents the 3-D bar plot of ρ versus E, K_s versus E, and ρ versus K_s for each dataset, and the evaluation is given by the overall accuracy with values between 0 and 1. The plot ρ versus E shows how the OA changes when the number of selected representative data points varies concerning the number of segments E. K_s versus E depicts how the OA is affected by the number of spectral segments and the kernel size used in the 2-D convolution to enhance the sparse coefficient matrix. Finally, ρ versus K_s shows the change in OA when the number of selected representative data points varies, and a specific kernel size is used in the 2-D convolution.

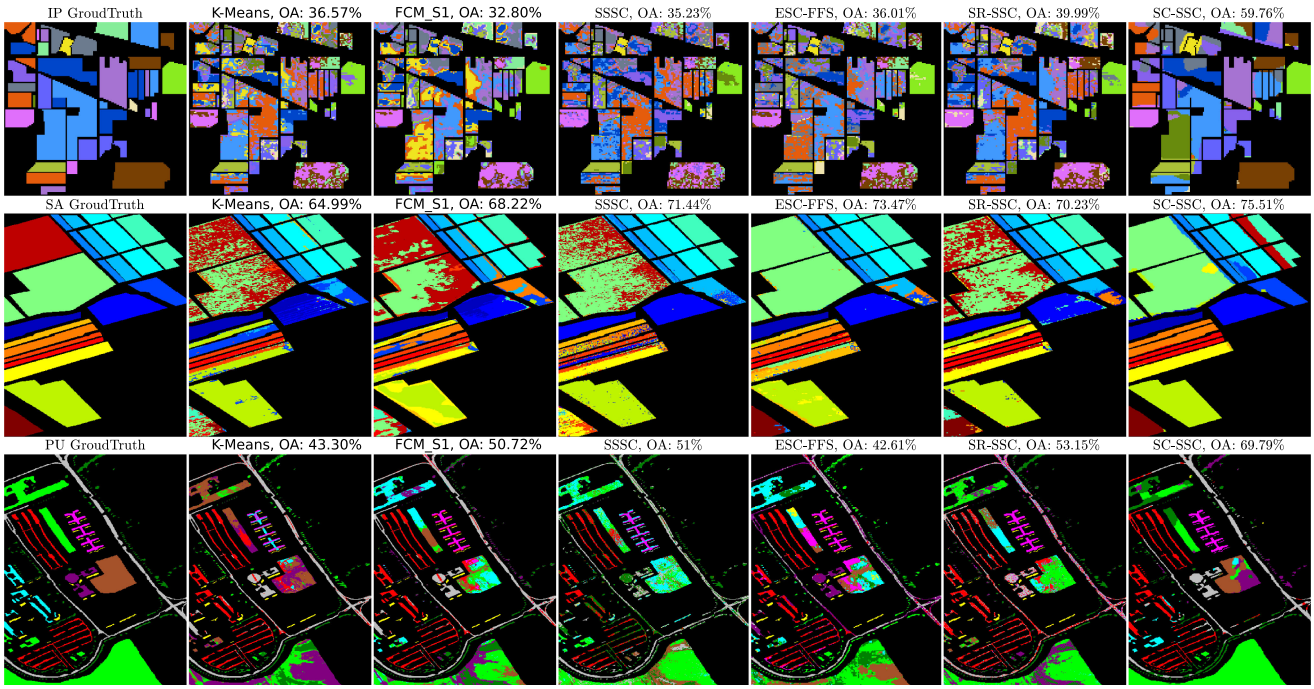


Fig. 7. Land cover maps on the Indian Pines (IP), Salinas Valley (SA), and University of Pavia (PU) Full images. The proposed method is compared with traditional clustering algorithms (K-Means and FCM_S1) and scalable SSC-based state-of-the-art methods.

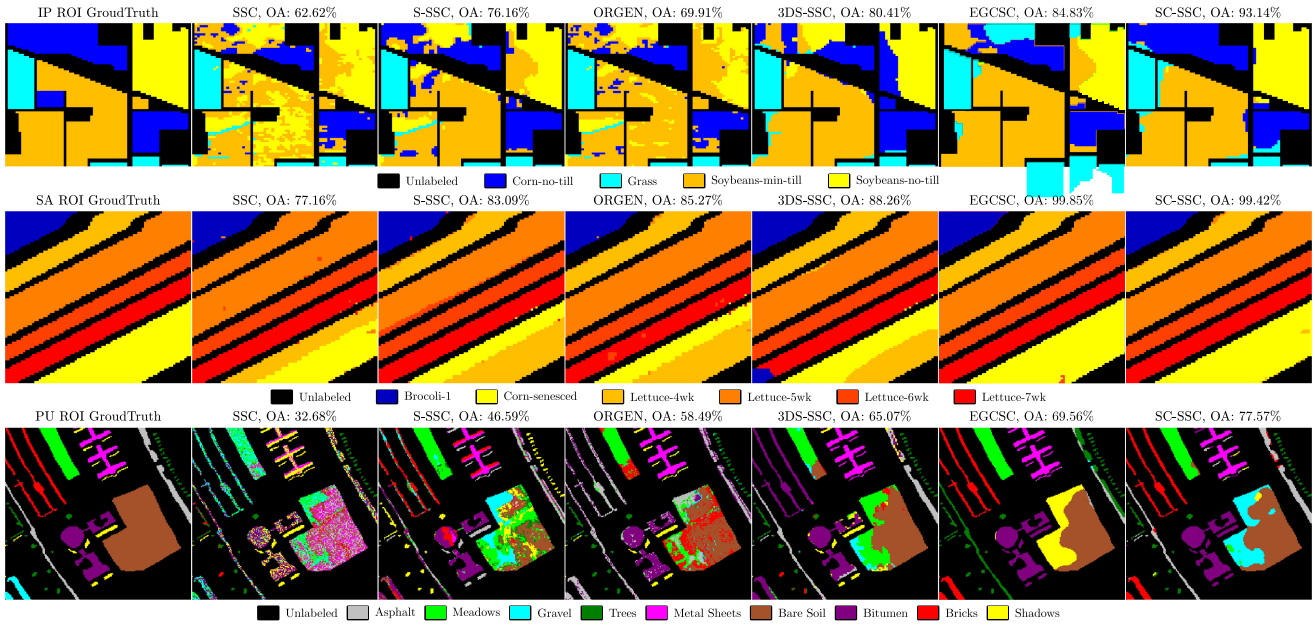


Fig. 8. Land cover maps of (first row) Indian Pines ROI, (second row) Salinas ROI, and (last row) the University of Pavia ROI. The proposed method is compared with nonscalable methods.

classified pixels by the total number of reference pixels. UA, AA, and OA values are presented in percentage, while Kappa coefficients and NMI values range from 0 (poor clustering) to 1 (perfect clustering). We also compare the methods in terms of clustering time.

B. Parameter Analysis and Tuning

In this section, we investigate the impact of the parameters ρ, E, and K_s in Algorithm 2. We conduct different experiments varying each parameter, with the others fixed. During simulations, we observe that the parameter ρ has a direct impact on the execution time of the proposed method. Fig. 5 presents the running time of SC-SSC for all the datasets. As shown, increasing ρ directly increases the running time; however, the most significant increment in time is given by the number of spectral pixels N, i.e., the size of the HSI, as observed with the differences in time between the curves. As we analyze in Section III-C2, this behavior is expected since the computational complexity of the algorithm is $O(\rho^2 N^3)$.

In our experiments, the parameters were varied between the following values: $\rho \in \{0.2, 0.25, 0.3, 0.35\}$, $E \in \{100, 300, 500, 700, 900, \ 1100, 1300, 1500, 1700, 1900\}$, and $K_s \in \{3, 5, 8, 16\}$. The parameter ρ determines the number of the selected most representative data points within each of the E segments, and K_s is the kernel size used in the 2-D convolution. Fig. 6 shows the performance of the proposed method with a different combination of the parameters for all the datasets, where the overall accuracy is shown between 0 and 1. By analyzing Fig. 6, we observe that the precision mainly changes with values of ρ and E. Also, $K_s = 8$ provided good results for the three images. In practice, an adequate balance when selecting the parameters ρ, E, and K_s is crucial to obtain the best performance. The selection of the best parameters often varies with the dataset's size; however, an effective way to select them is by using a grid search. Empirically, we found that the suitable values of ρ and E can be found in the range $[0.2, 0.3]$ and $[500, 1000]$, respectively. Furthermore, $\rho = 0.3$, $E = 700$, and $K_s = 8$ are good starting points. According to the empirical study, we provide a group of the best parameter setting in Table I.

TABLE I
SELECTED PARAMETERS IN ALGORITHM 2 FOR EACH HSI

Parameter	Indian Pines	Salinas	University of Pavia
ρ	0.35	0.2	0.3
E	1700	900	1900
K_s	8	3	8

TABLE II
ABLATION STUDY

				Indian Pines		Pavia		Salinas	
Experiment	PCA	Superpixels	2D Conv	OA	NMI	OA	NMI	OA	NMI
I	✓	✓		64.36	0.36	37.78	0.49	74.26	0.65
II		✓	✓	80.53	0.63	67.04	0.81	84.63	0.86
III			✓	41.72	0.29	38.11	0.46	43.33	0.42
IV	✓		✓	41.81	0.3	38.21	0.47	49.53	0.41
V		✓		65.84	0.37	47.60	0.49	73.29	0.77
VI	✓	✓	✓	**93.14**	**0.8**	**77.57**	**0.82**	**99.42**	**0.98**

The configuration shown in bold (Experiment VI) corresponds to our proposed approach.

C. Ablation Studies

We conduct six ablation experiments to investigate different configurations for the proposed workflow in Fig. 3. Specifically, we compare our proposed subspace clustering algorithm's performance when incorporating/excluding PCA, superpixels, and the 2-D convolution. Table II present the results obtained

from the different combinations in terms of OA and NMI for the three tested images. We observed that using superpixels to extract spatial similarities improves the clustering performance for the three tested images in all the cases, which evidence the importance of the neighboring spatial information in our workflow. Also, using superpixels and the 2-D convolution (Experiment II) leads to the second-best result, while only using 2-D convolution (Experiment III) does not lead to a significant clustering improvement. Finally, Experiment VI corresponds to our proposed approach, where we show that we achieve the best results in terms of OA and NMI when using the three operations, as described in the workflow in Fig. 3 or Algorithm 2.

D. Visual and Quantitative Results

1) Comparison With Scalable Methods: We compare the performance of SC-SSC with the scalable approaches: SSC-OMP, SSSC, ESC-FFS, and SR-SSC. Fig. 7 and Table III present the visual and quantitative evaluation results, respectively, on the full HSIs shown in Fig. 4. In the table, the best results are shown in bold, and the second-best is underlined. We observed that the proposed SC-SSC method outperforms the other approaches in terms of OA, Kappa, and NMI score from both qualitative and quantitative results. Although the proposed method is not the fastest one, it provides high clustering performance in a shorter amount of time than other methods.

2) Comparison With Nonscalable Methods: We present the obtained land cover maps on the Indian Pines, Salinas, and the University of Pavia ROIs in Fig. 8, where we compare the performance of our SC-SSC method with the nonscalable methods: SSC, S-SSC, ORGEN, and 3DS-SSC. The quantitative evaluations corresponding to the OA, Kappa, NMI, and Time with the nonscalable clustering methods are reported in Table IV. From Table IV, it can be observed that, in general, the proposed SC-SSC method performs better than others. Specifically, SC-SSC achieves an OA of 93.14% and 99.42%, in only 1.63 and 2.06 s, for the Indian Pines and Salinas dataset, respectively, which are remarkable results for unsupervised learning settings. Similarly, for the University of Pavia ROIs, it is observed from Table IV that the proposed SC-SSC achieves the best clustering performance in all the accuracy evaluation metrics, among all the other algorithms.

3) Comparison With Unsupervised-Deep-Learning-Based Methods: For the sake of completeness, we compare the proposed SC-SSC method with unsupervised-deep-learning-based methods based on autoencoders (AEs) for HSI clustering. Three of them were proposed in [45] (VAE, AE-GRU, and AE-LSTM), and the 3D-CAE method was proposed in [46], which is based on a 3-D convolutional AE. Note that we only compare our method with totally unsupervised deep learning approaches to make a fair comparison. Table V shows the quantitative results in terms of the NMI score. In the table, the best result is shown in bold font, and the second-best is underlined. As observed, our method obtains an NMI score of 0.601, 0.892, and 0.643 on Indian Pines, Salinas, and University of Pavia full HSIs, respectively, corresponding to the highest clustering scores.

TABLE III
CLUSTERING PERFORMANCE OF THE COMPARED METHODS ON THE INDIAN PINES, SALINAS, AND THE UNIVERSITY OF PAVIA DATASETS (FULL IMAGES)

Dataset	Class	SSC-OMP	SSSC	ESC-FFS	SR-SSC	SC-SSC
Indian Pines	Alfalfa	**7.69**	6.90	0.60	0.00	0.00
	Corn-no-till	37.04	34.33	57.79	47.95	**66.96**
	Corn-min-till	20.83	17.46	17.45	17.84	**55.25**
	Corn	22.73	15.56	11.34	8.85	**27.25**
	Grass-pasture	21.05	35.53	34.81	32.39	**90.52**
	Grass-trees	57.14	**84.91**	77.94	70.11	77.05
	Grass-mowed	**0.45**	0.00	0.29	0.00	0.00
	Hay-windrowed	25.81	88.29	85.91	76.91	**90.53**
	Oats	0.00	3.36	0.02	**4.36**	0.00
	Soybean-no-till	18.18	30.89	38.54	36.11	**64.15**
	Soybean-min-till	28.67	52.25	55.12	48.52	**62.23**
	Soybean-clean	6.32	22.17	17.44	15.76	**32.10**
	Wheat	8.24	37.60	37.53	56.35	**66.13**
	Woods	11.54	88.22	81.03	89.42	**91.49**
	Building-grass	5.56	17.33	25.51	22.15	**67.34**
	Stone-stell-towers	0.00	45.20	18.53	38.14	**49.73**
	AA	9.64	40.55	33.94	40.26	**56.37**
	OA	12.84	35.23	36.01	39.99	**59.76**
	Kappa	0.03	0.29	0.30	0.33	**0.55**
	NMI	0.03	0.42	0.43	0.44	**0.60**
	Time [s]	37.93	32.35	47.15	**16.39**	19.33
Salinas	Brocoli 1	11.34	0.00	**100.00**	0.00	88.39
	Brocoli 2	28.88	57.41	**99.68**	62.98	99.57
	Fallow	9.70	78.98	72.16	0.00	**80.25**
	Fallow Plow	7.43	91.56	89.52	**94.35**	64.12
	Fallow Smooth	22.58	58.11	71.98	71.82	**99.89**
	Stubble	21.54	99.05	99.70	**99.91**	97.04
	Celery	22.08	97.12	97.78	85.62	**100.00**
	Grapes	3.47	70.19	58.44	**70.68**	59.22
	Soil	24.68	91.18	**95.91**	85.58	91.11
	Corn	3.91	61.92	26.70	58.04	**89.95**
	Lettuce 4	3.97	**79.10**	24.01	0.00	0.00
	Lettuce 5	4.42	**81.09**	65.14	51.99	63.49
	Lettuce 6	1.90	**41.17**	0.00	0.00	0.00
	Lettuce 7	0.00	**58.92**	55.09	51.18	51.13
	Vineyard	11.57	**56.92**	25.00	49.22	0.00
	Vineyard trellis	3.05	0.00	98.53	98.89	**100.00**
	AA	10.95	64.76	73.15	61.57	**75.11**
	OA	10.74	71.44	73.47	70.23	**75.51**
	Kappa	0.05	0.68	0.70	0.67	**0.73**
	NMI	0.16	0.75	0.83	0.78	**0.85**
	Time [s]	78.18	**58.29**	615.40	168.18	580.33
University of Pavia	Asphalt	20.67	60.00	**95.60**	68.64	73.52
	Meadows	48.11	85.58	71.91	77.98	**96.81**
	Gravel	**13.73**	0.15	0.06	1.25	11.73
	Trees	9.54	31.28	18.03	**72.63**	32.48
	Metal sheets	81.71	**97.01**	36.74	60.15	86.22
	Bare soil	0.00	5.75	24.15	15.04	**97.53**
	Bitumen	0.00	4.26	**27.80**	0.00	0.00
	Bricks	0.00	55.27	**60.70**	39.11	54.54
	Shadows	0.00	43.32	62.99	**96.45**	0.00
	AA	16.48	39.76	**56.87**	53.50	52.23
	OA	35.34	51.00	42.61	53.15	**69.79**
	Kappa	0.07	0.39	0.33	0.42	**0.61**
	NMI	0.07	0.39	0.49	0.52	**0.64**
	Time [s]	201.52	**73.45**	1821.58	148.55	913.67

4) Comparison With Traditional Clustering Methods: We further compare our proposed approach with traditional centroid-based clustering methods, such as K-means [47], FCM [48], and FCM with spatial information (FCM_S1 and FCM_S2) [49]. Such methods are based on the fact that similar data points generate clusters in the feature space. These algorithms optimize the clusters by giving initial clustering centers and continuously updating their location until minimizing the sum of squared errors. Table VI shows the quantitative evaluation results. Furthermore, in Fig. 7, we show the clustering maps

TABLE IV
CLUSTERING PERFORMANCE OF THE COMPARED METHODS ON THE INDIAN PINES, SALINAS, AND THE UNIVERSITY OF PAVIA ROIS

Dataset	Class	SSC	S-SSC	ORGEN	3DS-SSC	EGCSC	SC-SSC
Indian Pines ROI	OA	62.62	76.16	69.91	80.41	84.83	**93.14**
	Kappa	0.48	0.66	0.56	0.72	0.64	**0.90**
	NMI	0.39	0.47	0.42	0.57	0.64	**0.79**
	Time [s]	285.57	301.41	668.36	341.21	69.37	**1.63**
Salinas ROI	OA	77.16	83.09	85.27	88.26	**99.85**	99.42
	Kappa	0.71	0.79	0.82	0.86	**0.99**	**0.99**
	NMI	0.85	0.84	0.87	0.87	0.99	**0.98**
	Time [s]	319.42	327.66	1355.79	377.11	92.95	**2.06**
University of Pavia ROI	OA	32.68	46.59	58.49	65.07	69.56	**77.57**
	Kappa	0.23	0.38	0.50	0.56	0.63	**0.72**
	NMI	0.41	0.49	0.59	0.65	**0.82**	**0.82**
	Time [s]	17821.25	10195.89	551.02	10501.29	171.21	**68.72**

TABLE V
QUANTITATIVE COMPARISON WITH UNSUPERVISED-DEEP-LEARNING-BASED METHODS IN TERMS OF NMI SCORE

Datasets \ Methods	VAE [45]	3D-CAE [46]	AE-GRU [45]	AE-LSTM [45]	SC-SSC
Indian Pines	0.429	0.504	0.515	0.478	**0.601**
Salinas	0.722	0.839	0.825	0.830	**0.892**
University of Pavia	0.505	0.639	0.524	0.569	**0.643**

TABLE VI
QUANTITATIVE COMPARISON WITH TRADITIONAL CLUSTERING ALGORITHMS IN TERMS OF OA

Datasets \ Methods	K-means [47]	FCM [48]	FCM_S1 [49]	FCM_S2 [49]	SC-SSC
Indian Pines	36.57	31.41	32.80	36.54	**59.76**
Salinas	64.99	55.93	68.22	64.38	**76.17**
Pavia University	43.30	47.98	50.72	49.60	**69.79**

obtained by the K-means and the FCM_S1 algorithms. Table VI and Fig. 7 show that such traditional clustering methods do not perform well on the real HSIs used in this work. In general, such methods are sensitive to noise and get easily stuck in a local optimum. On the other hand, our method is more robust to the noise present in the real HSI and achieves higher clustering performance.

V. CONCLUSION

In this work, we presented a new subspace clustering algorithm for unsupervised classification of HSIs, which is efficient and takes advantage of HSIs' neighboring spatial information to boost the clustering accuracy. Our method considers the spatial similarity among spectral pixels to select the most representative ones, such that all other adjacent points can be well represented by those pixels in terms of the sparse representation cost. Then, the obtained sparse coefficients matrix is enhanced by filtering the coefficients, and a fast spectral clustering algorithm gives the segmentation. Through simulations using traditional test HSIs, we demonstrated the effectiveness of our proposed method for fast HSI classification, obtaining remarkable high clustering performance compared with state-of-the-art SSC algorithms and even novel unsupervised-deep-learning-based methods.

REFERENCES

[1] G. A. Shaw and H. K. Burke, "Spectral imaging for remote sensing," *Lincoln Lab. J.*, vol. 14, no. 1, pp. 3–28, 2003.

[2] S. Liu, Q. Shi, and L. Zhang, "Few-shot hyperspectral image classification with unknown classes using multitask deep learning," *IEEE Trans. Geosci. Remote Sens.*, vol. 59, no. 6, pp. 5085–5102, Jun. 2021.

[3] Q. Shi, X. Tang, T. Yang, R. Liu, and L. Zhang, "Hyperspectral image denoising using a 3-D attention denoising network," *IEEE Trans. Geosci. Remote Sens.*, to be published, doi: 10.1109/TGRS.2020.3045273.

[4] Y. Lanthier, A. Bannari, D. Haboudane, J. R. Miller, and N. Tremblay, "Hyperspectral data segmentation and classification in precision agriculture: A multi-scale analysis," in *Proc. IEEE Int. Geosci. Remote Sens. Symp.*, 2008, vol. 2, pp. II– 585.

[5] P. S. Thenkabail and J. G. Lyon, *Hyperspectral Remote Sensing of Vegetation*. Boca Raton, FL, USA: CRC Press, 2016.

[6] D. He, Q. Shi, X. Liu, Y. Zhong, and X. Zhang, "Deep subpixel mapping based on semantic information modulated network for urban land use mapping," *IEEE Trans. Geosci. Remote Sens.*, to be published, doi: 10.1109/TGRS.2021.3050824.

[7] Q. Shi, M. Liu, S. Li, X. Liu, F. Wang, and L. Zhang, "A deeply supervised attention metric-based network and an open aerial image dataset for remote sensing change detection," *IEEE Trans. Geosci. Remote Sens.*, to be published, doi: 10.1109/TGRS.2021.3085870.

[8] B. Gessesse, W. Bewket, and A. Bräuning, "Model-based characterization and monitoring of runoff and soil erosion in response to land use/land cover changes in the Modjo Watershed, Ethiopia," *Land Degradation Develop.*, vol. 26, no. 7, pp. 711–724, 2015.

[9] M. Volpi and V. Ferrari, "Semantic segmentation of urban scenes by learning local class interactions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops*, 2015, pp. 1–9.

[10] X. Briottet *et al.*, "Military applications of hyperspectral imagery," *Proc. SPIE*, vol. 6239, 2006, Art. no. 62390B.

[11] P. Ghamisi *et al.*, "Advances in hyperspectral image and signal processing: A comprehensive overview of the state of the art," *IEEE Geosci. Remote Sens. Mag.*, vol. 5, no. 4, pp. 37–78, Dec. 2017.

[12] S. Li, W. Song, L. Fang, Y. Chen, P. Ghamisi, and J. A. Benediktsson, "Deep learning for hyperspectral image classification: An overview," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 9, pp. 6690–6709, Sep. 2019.

[13] K. Sanchez, C. Hinojosa, and H. Arguello, "Supervised spatio-spectral classification of fused images using superpixels," *Appl. Opt.*, vol. 58, no. 7, pp. B9–B18, 2019.

[14] C. Hinojosa, J. M. Ramirez, and H. Arguello, "Spectral-spatial classification from multi-sensor compressive measurements using superpixels," in *Proc. IEEE Int. Conf. Image Process.*, 2019, pp. 3143–3147.

[15] M. Paoletti, J. Haut, J. Plaza, and A. Plaza, "Deep learning classifiers for hyperspectral imaging: A review," *ISPRS J. Photogramm. Remote Sens.*, vol. 158, pp. 279–317, 2019.

[16] A. Kolesnikov, X. Zhai, and L. Beyer, "Revisiting self-supervised visual representation learning," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 1920–1929.

[17] E. Elhamifar and R. Vidal, "Sparse subspace clustering: Algorithm, theory, and applications," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 11, pp. 2765–2781, Nov. 2013.

[18] U. V. Luxburg, "A tutorial on spectral clustering," *Statist. Comput.*, vol. 17, no. 4, pp. 395–416, 2007.

[19] C. Hinojosa, F. Rojas, S. Castillo, and H. Arguello, "Hyperspectral image segmentation using 3D regularized subspace clustering model," *J. Appl. Remote Sens.*, vol. 15, 2021, Art. no. 016508.

[20] Y. Cai, Z. Zhang, Z. Cai, X. Liu, X. Jiang, and Q. Yan, "Graph convolutional subspace clustering: A robust subspace clustering framework for hyperspectral image," *IEEE Trans. Geosci. Remote Sens.*, vol. 59, no. 5, pp. 4191–4202, May 2021.

[21] C. Hinojosa, J. Bacca, and H. Arguello, "Coded aperture design for compressive spectral subspace clustering," *IEEE J. Sel. Topics Signal Process.*, vol. 12, no. 6, pp. 1589–1600, Dec. 2018.

[22] C. A. Hinojosa, J. Bacca, and H. Arguello, "Spectral imaging subspace clustering with 3-D spatial regularizer," in *Proc. Digit. Holography Three-Dimensional Imag. Conf.*, 2018, Art. no. JW 5E-7.

[23] H. Zhang, H. Zhai, L. Zhang, and P. Li, "Spectral–spatial sparse subspace clustering for hyperspectral remote sensing images," *IEEE Trans. Geosci. Remote Sens.*, vol. 54, no. 6, pp. 3672–3684, Jun. 2016.

[24] H. Zhai, H. Zhang, L. Zhang, P. Li, and A. Plaza, "A new sparse subspace clustering algorithm for hyperspectral remote sensing imagery," *IEEE Geosci. Remote Sens. Lett.*, vol. 14, no. 1, pp. 43–47, Jan. 2017.

[25] S. Huang, H. Zhang, and A. Pižurica, "Semisupervised sparse subspace clustering method with a joint sparsity constraint for hyperspectral remote sensing images," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 12, no. 3, pp. 989–999, Mar. 2019.
[26] X. Chen, W. Hong, F. Nie, D. He, M. Yang, and J. Z. Huang, "Spectral clustering of large-scale data by directly solving normalized cut," in *Proc. 24th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2018, pp. 1206–1215.
[27] B. Nasihatkon and R. Hartley, "Graph connectivity in sparse subspace clustering," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2011, pp. 2137–2144.
[28] X. Peng, L. Zhang, and Z. Yi, "Scalable sparse subspace clustering," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2013, pp. 430–437.
[29] C. You, D. Robinson, and R. Vidal, "Scalable sparse subspace clustering by orthogonal matching pursuit," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 3918–3927.
[30] J. A. Tropp and A. C. Gilbert, "Signal recovery from random measurements via orthogonal matching pursuit," *IEEE Trans. Inf. Theory*, vol. 53, no. 12, pp. 4655–4666, Dec. 2007.
[31] E. L. Dyer, A. C. Sankaranarayanan, and R. G. Baraniuk, "Greedy feature selection for subspace clustering," *J. Mach. Learn. Res.*, vol. 14, no. 1, pp. 2487–2517, 2013.
[32] Y. Chen, G. Li, and Y. Gu, "Active orthogonal matching pursuit for sparse subspace clustering," *IEEE Signal Process. Lett.*, vol. 25, no. 2, pp. 164–168, Feb. 2018.
[33] C. You, C.-G. Li, D. P. Robinson, and R. Vidal, "Oracle based active set algorithm for scalable elastic net subspace clustering," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 3928–3937.
[34] M. Abdolali, N. Gillis, and M. Rahmati, "Scalable and robust sparse subspace clustering using randomized clustering and multilayer graphs," *Signal Process.*, vol. 163, pp. 166–180, 2019.
[35] R. Tibshirani, "Regression shrinkage and selection via the LASSO," *J. Roy. Statist. Soc., Ser. B. (Methodol.)*, vol. 58, no. 1, pp. 267–288, 1996.
[36] C. You, C. Li, D. P. Robinson, and R. Vidal, "Scalable exemplar-based subspace clustering on class-imbalanced data," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 67–83.
[37] D. P. Williamson and D. B. Shmoys, *The Design of Approximation Algorithms*. Cambridge, U.K.: Cambridge Univ. Press, 2011.
[38] H. Zhai, H. Zhang, X. Xu, L. Zhang, and P. Li, "Kernel sparse subspace clustering with a spatial max pooling operation for hyperspectral remote sensing data interpretation," *Remote Sens.*, vol. 9, no. 4, 2017, Art. no. 335.
[39] J. Bacca, C. A. Hinojosa, and H. Arguello, "Kernel sparse subspace clustering with total variation denoising for hyperspectral remote sensing images," in *Proc. Math. Imag. Conf.*, 2017, Art. no. MTu 4C-5.
[40] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Süsstrunk, "SLIC superpixels compared to state-of-the-art superpixel methods," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 11, pp. 2274–2282, Nov. 2012.
[41] G. H. Golub and C. Reinsch, "Singular value decomposition and least squares solutions," in *Linear Algebra*. Berlin, Germany: Springer, 1971, pp. 134–151.
[42] X. Peng, Z. Yu, Z. Yi, and H. Tang, "Constructing the L2-graph for robust subspace learning and subspace clustering," *IEEE Trans. Cybern.*, vol. 47, no. 4, pp. 1053–1066, Apr. 2017.
[43] B. Efron *et al.*, "Least angle regression," *Ann. Statist.*, vol. 32, no. 2, pp. 407–499, 2004.
[44] J. Mairal, F. Bach, J. Ponce, and G. Sapiro, "Online learning for matrix factorization and sparse coding," *J. Mach. Learn. Res.*, vol. 11, no. Jan, pp. 19–60, 2010.
[45] L. Tulczyjew, M. Kawulok, and J. Nalepa, "Unsupervised feature learning using recurrent neural nets for segmenting hyperspectral images," *IEEE Geosci. Remote Sens. Lett.*, to be published, doi: 10.1109/LGRS.2020.3013205.
[46] J. Nalepa, M. Myller, Y. Imai, K.-I. Honda, T. Takeda, and M. Antoniak, "Unsupervised segmentation of hyperspectral images using 3-D convolutional autoencoders," *IEEE Geosci. Remote Sens. Lett.*, vol. 17, no. 11, pp. 1948–1952, Nov. 2020.
[47] S. Lloyd, "Least squares quantization in PCM," *IEEE Trans. Inf. Theory*, vol. 28, no. 2, pp. 129–137, Mar. 1982.
[48] J. C. Bezdek, *Pattern Recognition With Fuzzy Objective Function Algorithms*. Berlin, Germany: Springer, 2013.
[49] S. Chen and D. Zhang, "Robust image segmentation using FCM with spatial constraints based on new kernel-induced distance measure," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 34, no. 4, pp. 1907–1916, Aug. 2004.
[50] T. Lillesand, R. W. Kiefer, and J. Chipman, *Remote Sensing and Image Interpretation*. Hoboken, NJ, USA: Wiley, 2015.
[51] A. Strehl and J. Ghosh, "Cluster ensembles—A knowledge reuse framework for combining multiple partitions," *J. Mach. Learn. Res.*, vol. 3, pp. 583–617, 2002.

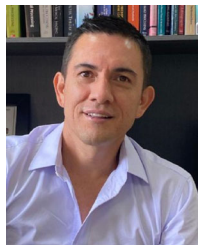
Carlos Hinojosa (Student Member, IEEE) received the B.Sc. and M.Sc. degrees in computer science in 2015 and 2018, respectively, from the Universidad Industrial de Santander, Bucaramanga, Colombia, where he currently working toward the Ph.D. degree in computer science.

His main research interests include computational imaging, optical code design, compressive imaging, computer vision, and sparse representation.

Esteban Vera (Member, IEEE) received the B.S. degree and Engineering diploma in electronics engineering in 1996 and 1999, respectively, and the M.S. and Ph.D. degrees in electrical engineering in 2003 and 2010, respectively, all from the Universidad de Concepción, Concepción, Chile.

From 2001 to 2007, he worked as an electronics engineer for different large telescope projects, first with the Paranal Observatory and then with the Gemini Observatory. In 2010, he was a Postdoctoral Research Associate with the University of Arizona, Tucson, AZ, USA. In 2013, he became a Research Scientist with Duke University, Durham, NC, USA. In 2016, he joined the School of Electrical Engineering, Pontificia Universidad Católica de Valparaíso, Valparaáso, Chile, where he is currently an Associate Professor. His research interests include computational imaging, compressed sensing, inverse problems, signal processing, machine learning, and the development of scientific and astronomical instrumentation.

Dr. Vera is a member of the International Society for Optical Engineers and a senior member of the Optical Society of America.



Henry Arguello (Senior Member, IEEE) received the B.Sc.Eng. degree in electrical engineering and the M.Sc. degree in electrical power from the Universidad Industrial de Santander, Bucaramanga, Colombia, in 2000 and 2003, respectively, and the Ph.D. degree in electrical engineering from the University of Delaware, Newark, DE, USA, in 2013.

He is currently a Professor with the Department of Systems Engineering, Universidad Industrial de Santander. In first semester in 2020, he was a Visiting Professor with Stanford University, Stanford, CA, USA, funded by Fulbright. His research interests include high-dimensional signal processing, computational imaging, compressed sensing, hyperspectral imaging, and deep optical sensor design.