

# Fast Three-Dimensional Empirical Mode Decomposition of Hyperspectral Images for Class-Oriented Multitask Learning

Zhi He, Jun Li, *Member, IEEE*, Lin Liu, Kai Liu, and Li Zhuo

**Abstract**—In this paper, we propose a fast 3-D empirical mode decomposition (fTEMD) method for hyperspectral images (HSIs) to achieve class-oriented multitask learning (cMTL). The major steps of the proposed method are twofold: 1) fTEMD and 2) cMTL. On the one hand, the traditional empirical mode decomposition is extended to its 3-D version, which naturally treats the HSI as a cube and effectively decomposes the HSI into several 3-D intrinsic mode functions (TIMFs). To accelerate the fTEMD, 3-D Delaunay triangulation is adopted to determine the distances of extrema, whereas separable filters are implemented to generate the envelopes. On the other hand, cMTL is performed on the TIMFs by taking those TIMFs as features of different tasks. The proposed cMTL learns the representation coefficients by taking advantage of the class labels and fully exploiting the information contained in each TIMF. Experiments conducted on three benchmark data sets demonstrate the effectiveness of the proposed method.

**Index Terms**—Classification, hyperspectral image (HSI), multi-task learning (MTL), three-dimensional empirical mode decomposition (TEMD).

## I. INTRODUCTION

A HYPERSPECTRAL image (HSI) [1] contains hundreds of consecutive narrowbands spanning the visible-to-infrared spectrum for the same area and, therefore, provides subtle discriminative information for various materials that can increase the potential to distinguish objects in the image area. Due to the ample information contained in the HSI, hyperspectral remote sensing has attracted extensive attention in many image processing topics, such as classification [2],

Manuscript received April 9, 2016; revised May 24, 2016; accepted June 18, 2016. Date of publication July 27, 2016; date of current version September 27, 2016. This work was supported in part by the National High Technology Research and Development Program of China (863 Program) under Grant 2013AA122302; by the National Natural Science Foundation of China under Grant 41501368, Grant 41522104, and Grant 41531178; and by the Fundamental Research Funds for the Central Universities under Grant 16lgpy04 and Grant 15lgjc24.

Z. He, J. Li, K. Liu, and L. Zhuo are with the Guangdong Provincial Key Laboratory of Urbanization and Geo-simulation, Center of Integrated Geographic Information Analysis, School of Geography and Planning, Sun Yat-sen University, Guangzhou 510275, China (e-mail: hezh8@mail.sysu.edu.cn; lijun48@mail.sysu.edu.cn; liuk6@mail.sysu.edu.cn; zhuoli@mail.sysu.edu.cn).

L. Liu is with the Guangdong Provincial Key Laboratory of Urbanization and Geo-simulation, Center of Integrated Geographic Information Analysis, School of Geography and Planning, Sun Yat-sen University, Guangzhou 510275, China, and also with the Department of Geography, University of Cincinnati, Cincinnati, OH 45221 USA (e-mail: liulin2@mail.sysu.edu.cn).

Color versions of one or more of the figures in this paper are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/TGRS.2016.2587672

fusion [3], unmixing [4], and target detection [5], among which classification is one of the fundamental task.

To identify meaningful features for classification, feature selection/extraction has been extensively investigated over the last few decades. Feature selection methods select a subset from a set of candidate features. Some widely used methods are based on a suboptimal search strategy [6], clustering [7]–[11], mutual information [12], [13], multitask sparsity pursuit [14], fractional-order Darwinian particle swarm optimization [15], and the firefly algorithm [16]. Feature extraction methods extract useful information from the given data. Several widespread approaches include principal component analysis [17], maximum noise fraction [18], wavelet transform (WT) [19], [20], 1-D empirical mode decomposition (1D-EMD) [21], 1-D singular spectrum analysis (1D-SSA) [22], and manifold-learning-inspired methods [23]–[25]. Notably that incorporating spatial information of neighboring pixels can yield significant performance, a number of recent techniques have been developed to extract spectral–spatial features, such as the extended morphological profile [26], [27], extended multiattribute profiles [28], 2-D EMD (2D-EMD) [29]–[31], and 2-D SSA (2D-SSA) [32]. In reality, an HSI data set is modeled as a 3-D cube, but the aforementioned methods are only based on vector or image representation. As such, many researchers have treated the HSI as a third-order tensor to identify the spectral–spatial features simultaneously. For instance, 3-D WT (3D-WT) is proposed for feature extraction methods in [33]–[35], the tensor discriminative locality alignment method is developed in [36] to remove redundant information of HSI, 3-D gray-level co-occurrence is presented in [37] to extract discriminant co-occurrence features, and the local tensor discriminative analysis technique [38] is proposed to integrate spectral–spatial feature extraction and tensor discriminant analysis.

Moreover, much work has been carried out to construct suitable classifiers [39]–[48] in the literature. Among available classifiers, the support vector machine (SVM) [39]–[41] and the sparse-representation-based classifier (SRC) [42]–[46] are two state-of-the-art works that have yielded impressive results. The SVM, which is insensitive to the curse of dimensionality, has been found to be remarkably effective in HSI classification. Motivated by the rapid development of compressed sensing, SRC has also emerged as a powerful tool for HSI classification and has become mainstream in recent years. Exploiting spatial information in the pixel-wise classifiers has attracted increasing research interests in improving the classification

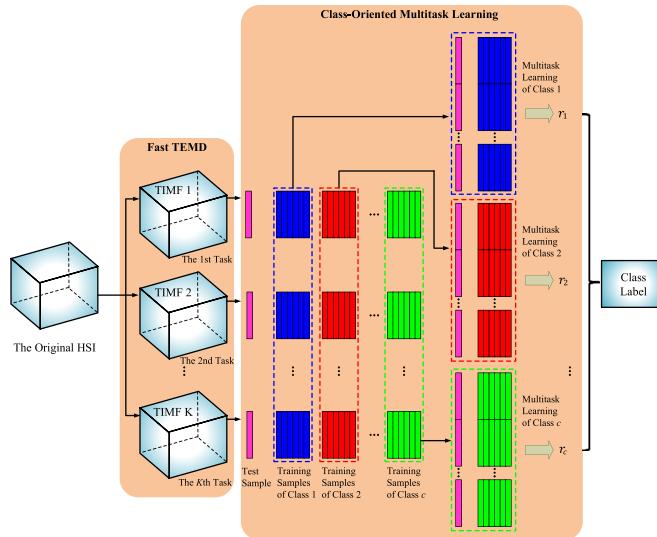


Fig. 1. Schematic illustration of the proposed fTEMd of HSI for cMTL.

performance. On the one hand, many refined versions of SVM have been proposed to exploit additional spatial correlation of neighboring pixels. For instance, a family of composite kernels is proposed in [49] to combine spectral and spatial information. Instead of assigning a specific kernel function, multiple kernel learning [50]–[53] is presented to encode local neighboring information of the scene by linearly combining various base kernels. In [54], Markov random field regularization is utilized to integrate spatial contextual information and refine the classification results of SVM. On the other hand, significant efforts have been concentrated on improving the SRC. For instance, the joint sparsity model (JSM) [55] is proposed to exploit the spatial dependence relations of neighboring pixels by assuming that the sparse codes associated with neighboring pixels share a common sparsity pattern. The weights of different neighboring pixels and the sparse representation coefficients of different training samples are simultaneously optimized by the nearest regularized joint sparse representation (NRJSR) classification [56]. In addition, many structured priors are proposed to incorporate additional spatial information, including Laplacian prior [57], low-rank prior [58], [59], group sparsity prior [60], and total variation prior [61]. Recently, multitask learning (MTL) [21], [62], [63] has been developed to exploit the joint sparsity prior of multiple features (e.g., spectral, texture, and shape). Note that MTL can improve the classification performance by identifying the intrinsic relationships of multiple tasks; it has received considerable attention in HSI classification. However, the label information is only adopted to calculate the residues but is ignored in learning the representation coefficients.

In this paper, a fast 3-D EMD (fTEMd)<sup>1</sup> is proposed for extracting the spectral–spatial features of HSI and achieving class-oriented MTL (cMTL) for HSI classification. Fig. 1 gives a schematic illustration of the proposed HSI classification meth-

od, which contains two major steps: 1) fTEMd and 2) cMTL. In the first step, the original HSI is decomposed into several 3-D intrinsic mode functions (TIMFs) by fTEMd, which is a fast 3-D extension of the traditional 1D/2D-EMD. Note that the computational complexity and memory requirements will exponentially increase if the number of dimensions increases; it is challenging to extend the 1D/2D-EMD to its 3-D version. Although multidimensional ensemble EMD (EEMD) is proposed in [64] to decompose multidimensional data by applying the EEMD in the first, second, and third dimensions, respectively, it involves heavy calculation burden and is unable to treat the 3-D cube as a whole entity. Recently, a fast and adaptive tridimensional EMD (FATEMD) [65] has been developed to generate upper/lower envelopes by order statistics filters and naturally decompose a volume into a number of components. Versatile as FATEMD is, its efficiency should be further improved. As stated in [65], more than 4 h are required to decompose a  $256 \times 256 \times 35$  brain volume. To accelerate the proposed fTEMd, 3-D Delaunay triangulation (3D-DT) is applied to calculate the distances among extrema, whereas separable filters are adopted to generate the envelopes. In the second step, cMTL is performed on the TIMFs by taking each of the TIMFs as a specific task. Different from the existing MTL, cMTL learns the representation coefficients class by class and determines the class label of a test sample by a minimum error of the class-oriented representation.

Compared with the hyperspectral classification literature, the contributions and novel aspects of the proposed method can be summarized as follows.

- 1) fTEMd is proposed to treat the HSI as a 3-D cube and decompose the HSI into several TIMFs. Compared with the vector/image-based methods, the proposed fTEMd method is capable of maintaining the structural information and obtaining the spectral–spatial features<sup>2</sup> of the original HSI.
- 2) cMTL is proposed to achieve class-oriented representation and significant classification of the test samples. Compared with the existing MTL, the proposed cMTL learns the representation coefficients by taking advantage of the class labels and fully exploiting the class-oriented information of the TIMFs.

The layout of this paper is as follows. Section II describes the proposed fTEMd method. Section III presents the cMTL for HSI classification. Experimental results on three benchmark HSIs are illustrated in Section IV. Finally, conclusions are drawn in Section V.

<sup>1</sup>As will be explained in the experiments, the reasons why fTEMd is fast lie in that: 1) the fTEMd is much faster than the existing FATEMD [65]; 2) the fTEMd is also much faster than the existing 1D-EMD and 2D-EMD.

<sup>2</sup>An HSI data set is intrinsically formed as a 3-D cube that contains a spectral dimension and two spatial dimensions. The vector/image-based methods are only focused on the spectral profile of each pixel or a single spectral band that has two spatial dimensions. Different from the vector/image-based methods, the proposed fTEMd respects the 3-D nature of the HSI and treats the 3-D cube as a whole to extract the spectral and spatial information simultaneously. As will be shown in Section II, the fTEMd exploits the correlation along the wavelength axis and spatial axes. Therefore, the spatial information can be implicitly included within the TIMFs.

## II. fTEMD-BASED FEATURE DESCRIPTOR

Over the past few years, EMD [66] has been of growing interest in a broad range of applications for extracting local oscillations and time-frequency distribution from the nonlinear/nonstationary data. Contrary to Fourier or WT, EMD is direct, intuitive, and adaptive, with an unpredetermined basis function derived from the data. Since the HSI acquired by hyperspectral sensors is a data cube, it is suitable to analyze the HSI by the 3-D version of EMD. However, the FATEMD proposed in [65] is time consuming, with much of the time spent on determining the filter sizes and on producing the mean envelopes. In this section, we propose the fTEMD to speed up the existing FATEMD.

### A. Framework of the fTEMD

The fTEMD is realized by iteratively detecting the local maxima/minima of the cube, generating upper/lower envelopes of the extrema by MAX/MIN filters, subtracting the smoothed mean envelopes to isolate the high-frequency oscillatory components termed as TIMFs, and repeating the aforementioned iteration recursively on the remaining data. Specifically, suppose the HSI cube is modeled as  $\mathbf{V}(m, n, b)$  with  $m, n, b \in \mathbb{N}^+$  representing the data size, the fTEMD can be implemented by the following iterative sifting process.

- 1) Let  $i = 1$ ,  $\mathbf{R}_i(m, n, b) = \mathbf{V}(m, n, b)$ .
- 2) Set  $j = 1$ ,  $\mathbf{h}_{i,j-1}(m, n, b) = \mathbf{R}_i(m, n, b)$ .
- 3) Generate the maps of local maxima and minima denoted as  $\mathbf{M}_{\max}(m, n, b)$  and  $\mathbf{M}_{\min}(m, n, b)$ , respectively, by browsing  $\mathbf{h}_{i,j-1}(m, n, b)$  with a  $3 \times 3 \times 3$  neighboring window. A local maximum (or minimum) is a particular point that is strictly greater (or smaller) than its neighborhoods.
- 4) Determine the size of filters (i.e., MAX/MIN filter and average filter) that will be used to create the upper/lower envelopes of the extrema in step 5) and the smoothed mean envelopes in step 6). As will be shown in Section II-B, the filter size can be effectively calculated by the 3D-DT.
- 5) Create the upper envelope  $\mathbf{E}_{\max}(m, n, b)$  of the maxima by the MAX filter and the lower envelope  $\mathbf{E}_{\min}(m, n, b)$  of the minima by the MIN filter and, subsequently, generate the mean envelope  $\mathbf{M}(m, n, b)$  by

$$\mathbf{M}(m, n, b) = \frac{\mathbf{E}_{\max}(m, n, b) + \mathbf{E}_{\min}(m, n, b)}{2}. \quad (1)$$

- 6) Calculate the smoothed mean envelope  $\mathbf{M}_s(m, n, b)$  by the average filter. As will be explained in Section II-C, all of the aforementioned filters (i.e., MAX/MIN filter and average filter) are separable, which accelerate the fTEMD algorithm by performing a 1-D filter three times to obtain the same results of the original complicated 3-D filter.
- 7) Update the  $j$ th medium variable  $\mathbf{h}_{i,j}(m, n, b)$  obtained in extracting the  $i$ th TIMF by subtracting

the smoothed envelope, i.e.,  $\mathbf{h}_{i,j}(m, n, b) = \mathbf{h}_{i,j-1}(m, n, b) - \mathbf{M}_s(m, n, b)$ .

- 8) Check whether  $\mathbf{h}_{i,j}(m, n, b)$  is a TIMF. The stop criterion for determining whether  $\mathbf{h}_{i,j}(m, n, b)$  is a TIMF can be expressed as

$$SD_{i,j} = \frac{\sum_{x=1}^m \sum_{y=1}^n \sum_{z=1}^b |\mathbf{h}_{i,j}(x, y, z) - \mathbf{h}_{i,j-1}(x, y, z)|^2}{\sum_{x=1}^m \sum_{y=1}^n \sum_{z=1}^b |\mathbf{h}_{i,j-1}(x, y, z)|^2} \quad (2)$$

where  $(x, y, z)$  denotes the coordinate of the 3-D cube, and  $SD_{i,j}$  refers to the stop criterion of the  $j$ th iteration in obtaining the  $i$ th TIMF. If  $SD_{i,j}$  is low (e.g., lower than 0.05), the medium variable  $\mathbf{h}_{i,j}(m, n, b)$  can be taken as a TIMF, i.e., let  $\mathbf{C}_i(m, n, b) = \mathbf{h}_{i,j}(m, n, b)$ ; otherwise, let  $j = j + 1$  and go back to step 3).

- 9) Update the residue  $\mathbf{R}_{i+1}(m, n, b) = \mathbf{R}_i(m, n, b) - \mathbf{C}_i(m, n, b)$ .
- 10) Check whether  $\mathbf{R}_{i+1}(m, n, b)$  contains less than three extrema. If so, let  $\mathbf{R}_{i+1}(m, n, b)$  be the final residue and complete the whole decomposition; otherwise, let  $i = i + 1$  and go back to step 2).

In terms of the aforementioned steps 1)–10), the HSI data set  $\mathbf{V}(m, n, b)$  can be decomposed as

$$\mathbf{V}(m, n, b) = \sum_{i=1}^{I+1} \mathbf{C}_i(m, n, b) \quad (3)$$

where  $\mathbf{C}_i(m, n, b)$  denotes the  $i$ th ( $i = 1, 2, \dots, I$ ) TIMF, and the final residue  $\mathbf{R}_{I+1}(m, n, b)$  is consistently represented as  $\mathbf{C}_{I+1}(m, n, b)$  for simplicity.

Moreover, the flowchart of the proposed fTEMD algorithm is depicted in Fig. 2, in which the steps shown in the blue dashed box are the main differences between fTEMD and the existing FATEMD. As will be demonstrated in Section II-B and C, the filter size and envelopes can be determined by the 3D-DT and separable filters, respectively.

### B. Determination of the Filter Size

As mentioned in step 5) of the fTEMD, MAX/MIN filters are adopted to generate the upper/lower envelopes rather than utilizing the 3-D scatter data interpolation. As such, it is of great importance to determine the size of filters in step 4). In [65], the distances among all maxima (or minima) should be calculated to gain the distance vector  $\mathbf{d}_{adj-\max}$  (or  $\mathbf{d}_{adj-\min}$ ), where  $\mathbf{d}_{adj-\max}$  (or  $\mathbf{d}_{adj-\min}$ ) is produced by sorting the array of the nearest Euclidean distances among each maximum (or minimum) and the other maxima (or minima). However, we observe that only the nearest Euclidean distances are sorted in  $\mathbf{d}_{adj-\max}$  (or  $\mathbf{d}_{adj-\min}$ ) while most of the other distances are redundant, particularly those maxima (or minima) far away from the particular maximum (or minimum). It is apparent that the computational cost will be reduced if the amount of distance calculation decreases.

In this paper, 3D-DT is applied to recognize the nearest extrema of a certain extremum. 3D-DT is a triangulation of 3-D

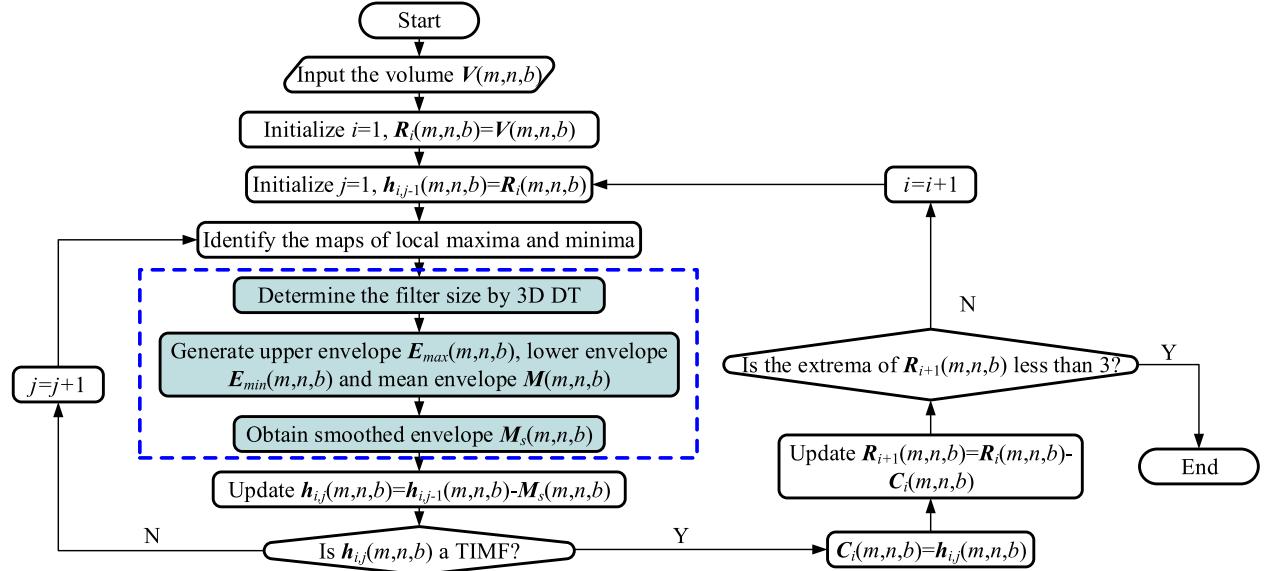


Fig. 2. Flowchart of the proposed fTEMD algorithm. Steps in the blue dashed box demonstrate the main differences between fTEMD and the existing FATEMD.

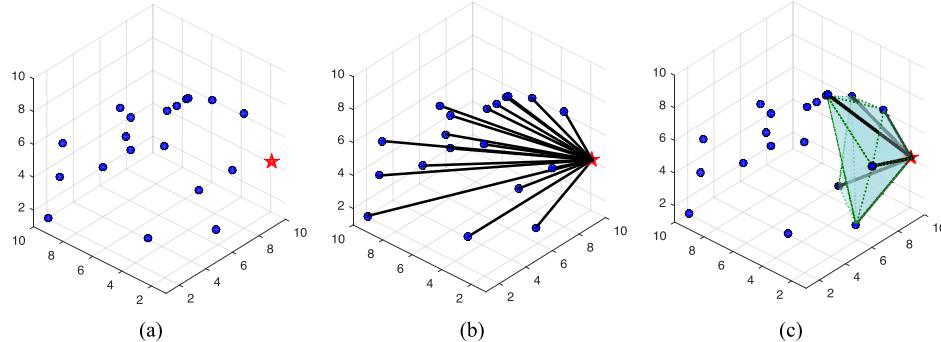


Fig. 3. Schematic comparison of two nearest Euclidean distance identification methods. “★” denotes a certain maximum, and “●” denotes the other maxima. (a) Map of local maxima  $M_{\max}(m, n, b)$  that contains 20 local maxima. (b) In FATEMD, all Euclidean distances (19 bold black lines) among a certain maximum and the other maxima should be calculated. (c) In the proposed fTEMD, only the Euclidean distances (six bold black lines) among a certain maximum and its neighbors are taken into account.

scatter data such that no point is inside the circumsphere of each tetrahedron. As to the fTEMD, the maxima (or minima) can be treated as 3-D scatter data, and 3D-DT can be constructed in  $M_{\max}(m, n, b)$  (or  $M_{\min}(m, n, b)$ ). In this regard, the nearest distance of a maximum and the other maxima is obtained by the nearest distance among the particular maximum and its neighbors located in the same tetrahedron. The nearest distance of minima can also be analogously detected.

As an example, Fig. 3 schematically compares the aforementioned two nearest Euclidean distance identification methods. Fig. 3(a) plots a map of local maxima  $M_{\max}(m, n, b)$  that contains 20 local maxima; the red pentagram denotes a certain maximum, and the blue dots represent the other maxima. Fig. 3(b) demonstrates that all Euclidean distances among the certain maximum and other 19 maxima should be calculated in FATEMD, whereas Fig. 3(c) indicates that only the neighbors detected by 3D-DT are involved in calculating the Euclidean distances in the proposed fTEMD. It is worth stressing that although the calculation of Euclidean distances in FATEMD can be accelerated by multithread or parallel processing, the fTEMD is still faster than the FATEMD if the same acceleration strategy (e.g., multithread or parallel processing) is adopted in

both methods. This is due to the fact that fTEMD theoretically decreases the amount of distance information that should be calculated. In practical applications (e.g., HSI), the advantage of fTEMD will become more significant with a larger number of local maxima (or minima).

Moreover, let  $d_1$ ,  $d_2$ ,  $d_3$ , and  $d_4$  denote the four types of distances defined as

$$d_1 = \min \{\min\{d_{adj-\max}\}, \min\{d_{adj-\min}\}\} \quad (4)$$

$$d_2 = \max \{\min\{d_{adj-\max}\}, \min\{d_{adj-\min}\}\} \quad (5)$$

$$d_3 = \min \{\max\{d_{adj-\max}\}, \max\{d_{adj-\min}\}\} \quad (6)$$

$$d_4 = \max \{\max\{d_{adj-\max}\}, \max\{d_{adj-\min}\}\}. \quad (7)$$

Instead of choosing a particular type of distance (i.e.,  $d_1$ ,  $d_2$ ,  $d_3$ , or  $d_4$ ), the filter size  $d$  in this paper is obtained by the mean of four types of distances, i.e.,

$$d = \left[ \frac{d_1 + d_2 + d_3 + d_4}{4} \right]_{odd} \quad (8)$$

where  $[.]_{odd}$  refers to the nearest odd integer operator, and  $d_1$ ,  $d_2$ ,  $d_3$ , and  $d_4$  are the four types of distances displayed in (4)–(7). The reason for taking the mean of four types of

distances (i.e.,  $d_1, d_2, d_3$ , and  $d_4$ ) as filter size  $d$  is that the mean can mitigate the influence of a particular distance and, thus, yields more robust results than any particular type of distances.

### C. Generation of Envelopes

Having determined filter size  $d$ , the upper envelope and the lower envelope can be generated, respectively, as

$$\mathbf{E}_{\max}(x, y, z) = \text{MAX}\{\mathbf{h}_{i,j-1}(s, t, u)\} \quad (9)$$

$$\mathbf{E}_{\min}(x, y, z) = \text{MIN}\{\mathbf{h}_{i,j-1}(s, t, u)\} \quad (10)$$

where  $s \in [x - \tilde{d}, x + \tilde{d}], t \in [y - \tilde{d}, y + \tilde{d}], u \in [z - \tilde{d}, z + \tilde{d}]$ ,  $\tilde{d} = ((d - 1)/2)$ , and  $\mathbf{h}_{i,j-1}$  is the  $(j - 1)$ th medium variable obtained in extracting the  $i$ th TIMF, and  $\text{MAX}\{\cdot\}$  (or  $\text{MIN}\{\cdot\}$ ) is the MAX (or MIN) filter, which is specifically defined as the maximum (or minimum) of all data points located in the  $d \times d \times d$  region of  $\mathbf{h}_{i,j-1}$ .

Subsequently, (1) is adopted to compute the mean envelope  $\mathbf{M}(m, n, b)$ , which can then be smoothed by the following average filter:

$$\mathbf{M}_s(x, y, z) = \frac{1}{d \times d \times d} \sum_{s=x-\tilde{d}}^{x+\tilde{d}} \sum_{t=y-\tilde{d}}^{y+\tilde{d}} \sum_{u=z-\tilde{d}}^{z+\tilde{d}} \mathbf{M}(s, t, u) \quad (11)$$

where  $(x, y, z)$  denotes the coordinate of an arbitrary point in the smoothed envelope  $\mathbf{M}_s(m, n, b)$ .

It is worth underlining that the upper/lower envelopes and the smoothed mean envelope shown in (9)–(11) can be effectively calculated by separable filters. On the one hand, the MAX and MIN filters have separability property [67], which facilitates the implementation of the MAX (or MIN) filter along the rows and columns of a 2-D image independently. As to the fTEMD, we can generalize the separability property of MAX and MIN filters to a 3-D scenario. In other words, rather than complicated computations in the 3-D domain, the MAX (or MIN) filter in (9) [or (10)] is simply realized by performing a 1-D MAX (or MIN) filter along the first, second, and third dimensions of the medium variable  $\mathbf{h}_{i,j-1}$ . On the other hand, the average filter in (11) is also a separable linear filter that can be achieved by separable convolution. In greater detail, (11) can be rewritten as

$$\begin{aligned} \mathbf{M}_s(x, y, z) &= \sum_{s=x-\tilde{d}}^{x+\tilde{d}} \sum_{t=y-\tilde{d}}^{y+\tilde{d}} \sum_{u=z-\tilde{d}}^{z+\tilde{d}} \mathbf{M}(s, t, u) \tilde{\mathbf{K}} \\ &\quad \times (x - s, y - t, z - u) \\ &= (\mathbf{M} * \tilde{\mathbf{K}})(x, y, z) \end{aligned} \quad (12)$$

where “\*” indicates the convolution operator,  $\tilde{\mathbf{K}} = (\mathbf{K}/(d \times d \times d))$  denotes a 3-D convolution kernel, and  $\mathbf{K}$  is an all-ones 3-D matrix.

The 3-D matrix  $\mathbf{K}$  can be treated as a third-order tensor, whose CANDECOMP/PARAFAC (CP) decomposition is

$$\mathbf{K} = \mathbf{a} \circ \mathbf{b} \circ \mathbf{c} \quad (13)$$

where “ $\circ$ ” indicates the outer product, and  $\mathbf{a}$ ,  $\mathbf{b}$ , and  $\mathbf{c}$  are all-ones vectors, all of whose lengths are equal to  $d$ .

We hereby state that  $\mathbf{K}$  is separable since the rank of the Kruskal form of  $\mathbf{K}$  shown in (13) is equal to 1. In this regard, the average filter in (11) can also be significantly realized by first performing a first-dimensional 1-D convolution, followed by a second-dimensional 1-D convolution, and finally, a third-dimensional 1-D convolution. In merit of performing the 1-D convolutions three times, we obtain the same results as applying the original 3-D convolution but with reduced computational complexity from the order of  $d^3$  to  $3d$  per pixel. As to an HSI cube  $\mathbf{V}(m, n, b)$  with  $m, n, b \in \mathbb{N}^+$  denoting the data size, the computational complexity can be reduced from  $d^3 mnb$  to  $3dmnb$  for the whole data cube. Moreover, notably that the main computational cost of fTEMD is filter size determination and envelope creation, the computational demand of fTEMD is  $o(\sum_{i=1}^I \sum_{j=1}^{S_i} [6\hat{M}_{i,j} + 6\check{M}_{i,j} + 9d_{i,j}mnb])$ , where  $I$  is the total number of TIMFs,  $S_i$  is the number of sifting iterations in extracting the  $i$ th TIMF, and  $\hat{M}_{i,j}$ ,  $\check{M}_{i,j}$ , and  $d_{i,j}$  denote the number of maxima and minima and the filter size generated in the  $j$ th sifting iteration to extract the  $i$ th TIMF, respectively.

### III. CLASS-ORIENTED MULTITASK LEARNING

The components  $\mathbf{C}_i(m, n, b)$ ,  $i = 1, 2, \dots, I + 1$  extracted by fTEMD can be taken as different features of the HSI cube. To make full use of those features, a cMTL-based classifier is proposed in this section. Since each component extracted by fTEMD contains different but useful spectral–spatial features of the original HSI data set, we take all of the components  $\mathbf{C}_i(m, n, b)$ ,  $i = 1, 2, \dots, I + 1$  as tasks of the cMTL (see Fig. 1). Let  $K = I + 1$ , there are  $K$  tasks in the cMTL. Moreover, we demonstrate  $\mathbf{D}^k = [\mathbf{D}_1^k, \mathbf{D}_2^k, \dots, \mathbf{D}_J^k]$  as the training feature matrix of the  $k$ th ( $k = 1, 2, \dots, K$ ) task, where  $\mathbf{D}_j^k \in \mathbb{R}^{m_k \times p_j}$ ,  $j = 1, 2, \dots, J$  is associated with the  $j$ th class,  $m_k$  represents the length of the  $k$ th task,  $p_j$  denotes the number of training samples in class  $j$ , and  $\sum_{j=1}^J p_j = p$  is the total number of training samples. Given an unknown test sample  $\mathbf{y}$ , suppose the  $k$ th task of  $\mathbf{y}$  can be expressed by the linear representation of the  $j$ th class, i.e.,

$$\mathbf{y}^k = \mathbf{D}_j^k \mathbf{w}_j^k + \varepsilon_j^k, \quad k = 1, 2, \dots, K, \quad j = 1, 2, \dots, J \quad (14)$$

where  $\mathbf{y}^k \in \mathbb{R}^{m_k \times 1}$  denotes the  $k$ th task of  $\mathbf{y}$ ,  $\mathbf{w}_j^k \in \mathbb{R}^{p_j \times 1}$  refers to the coefficient vector of the  $k$ th task in the  $j$ th class, and  $\varepsilon_j^k \in \mathbb{R}^{m_k \times 1}$  denotes the reconstruction error of the  $k$ th task in the  $j$ th class.

Let  $\mathbf{w}_j = [\mathbf{w}_j^1, \mathbf{w}_j^2, \dots, \mathbf{w}_j^K]$  be the coefficient vector of the  $j$ th class across different tasks,  $\mathbf{w}_j$  can then be obtained by the following multitask least squares regression:

$$\mathbf{w}_j = \arg \min_{\mathbf{w}_j} \frac{1}{2} \sum_{k=1}^K \|\mathbf{y}^k - \mathbf{D}_j^k \mathbf{w}_j^k\|_2^2 + \lambda \|\mathbf{w}_j\|_2, \quad j = 1, 2, \dots, J. \quad (15)$$

It is notable that (15) is a special case of the existing MTL method [21], [63], [68] with a single class, which can then be effectively solved by the accelerated proximal gradient (APG).

In greater detail, APG alternately updates a weight matrix sequence  $\{\mathbf{w}_j^{(i)} = [\mathbf{w}_j^{k,(i)}]\}_{i \geq 1}$  and an aggregation matrix sequence  $\{\mathbf{v}_j^{(i)} = [\mathbf{v}_j^{k,(i)}]\}_{i \geq 1}$ . The main update rules of each iteration can be summarized as follows.

- The generalized gradient mapping step to update  $\mathbf{w}_j^{(i+1)}$  with the current  $\mathbf{v}_j^{(i)}$ . Thus
  - $\mathbf{w}_j^{k,(i+(1/2))} = \mathbf{v}_j^{k,(i)} - \mu[-(\mathbf{D}_j^k)^T \mathbf{y}^k + (\mathbf{D}_j^k)^T \mathbf{D}_j^k \mathbf{v}_j^{k,(i)}]$ ,  
 $k = 1, 2, \dots, K$
  - $\mathbf{w}_j^{(i+1)} = [1 - (\lambda\mu/\|\mathbf{w}_j^{(i+(1/2))}\|_2)]_+ \mathbf{w}_j^{(i+(1/2))}$ .
- The aggregation forward step to update  $\mathbf{v}_j^{(i+1)}$  by the linear combination of  $\mathbf{w}_j^{(i)}$  and  $\mathbf{w}_j^{(i+1)}$ . Thus
  - $\alpha^{(i+1)} = (2/(i+3))$
  - $\mathbf{v}_j^{k,(i+1)} = \mathbf{w}_j^{k,(i+1)} + ((1-\alpha^{(i)})\alpha^{(i+1)}/\alpha^{(i)})$   
 $(\mathbf{w}_j^{k,(i+1)} - \mathbf{w}_j^{k,(i)})$ ,  $k = 1, 2, \dots, K$

where  $\mu$  is the step size parameter, and  $[.]_+ = \max\{0, \cdot\}$ .

In APG, the regularization parameter  $\lambda$  and the number of iterations are set the same for each class. After solving  $\mathbf{w}_j$  for each  $j$ , the class label of the test sample  $\mathbf{y}$  yields

$$\text{Class } \mathbf{y} = \arg \min_j r_j \quad (16)$$

where  $r_j = \sum_{k=1}^K \|\mathbf{y}^k - \mathbf{D}_j^k \mathbf{w}_j^k\|_2^2$ ,  $j = 1, 2, \dots, J$ . The lowest-reconstruction-error rule of (16) indicates that the test sample  $\mathbf{y}$  that belongs to a certain class (e.g., class  $j$ ) can be more accurately represented by the training samples from class  $j$  than other classes.

Note that inner products are involved in the update steps of the aforementioned APG method; we can straightforwardly extend the APG to its kernel version. This means that a nonlinear function  $\phi^k$  for each task  $k$  can be introduced to map the training and test samples from the original space to the high-dimensional feature space, in which  $\phi^k(\mathbf{D}_j^k)^T \phi^k(\mathbf{D}_j^k) = \text{Ker}(\mathbf{D}_j^k, \mathbf{D}_j^k)$ ,  $\phi^k(\mathbf{D}_j^k)^T \phi^k(\mathbf{y}^k) = \text{Ker}(\mathbf{D}_j^k, \mathbf{y}^k)$ , and “Ker” denotes the kernel function. As such, (15) can be rewritten as

$$\begin{aligned} \mathbf{w}_j = \arg \min_{\mathbf{w}_j} \frac{1}{2} \sum_{k=1}^K \|\phi^k(\mathbf{y}^k) - \phi^k(\mathbf{D}_j^k) \mathbf{w}_j^k\|_2^2 \\ + \lambda \|\mathbf{w}_j\|_2, \quad j = 1, 2, \dots, J. \end{aligned} \quad (17)$$

Accordingly, the first substep in the generalized gradient mapping can be changed to

$$\begin{aligned} \mathbf{w}_j^{k,(i+\frac{1}{2})} &= \mathbf{v}_j^{k,(i)} - \mu \\ &\times [-\text{Ker}(\mathbf{D}_j^k, \mathbf{y}^k) + \text{Ker}(\mathbf{D}_j^k, \mathbf{D}_j^k) \mathbf{v}_j^{k,(i)}], \quad k = 1, 2, \dots, K. \end{aligned} \quad (18)$$

Moreover, the rest of the steps of the APG do not need to be changed, and the classification decision rule in (16) can be

replaced by

$$\begin{aligned} \text{Class } \mathbf{y} &= \arg \min_j r_j \\ &= \arg \min_j \sum_{k=1}^K \|\phi^k(\mathbf{y}^k) - \phi^k(\mathbf{D}_j^k) \mathbf{w}_j^k\|_2^2 \\ &= \arg \min_j \sum_{k=1}^K \left[ (\phi^k(\mathbf{y}^k))^T \phi^k(\mathbf{y}^k) \right. \\ &\quad \left. - 2 (\phi^k(\mathbf{D}_j^k))^T \phi^k(\mathbf{y}^k) \mathbf{w}_j^k \right. \\ &\quad \left. + (\mathbf{w}_j^k)^T (\phi^k(\mathbf{D}_j^k))^T \phi^k(\mathbf{D}_j^k) \mathbf{w}_j^k \right] \\ &= \arg \min_j \sum_{k=1}^K \left[ -2 (\phi^k(\mathbf{D}_j^k))^T \phi^k(\mathbf{y}^k) \mathbf{w}_j^k \right. \\ &\quad \left. + (\mathbf{w}_j^k)^T (\phi^k(\mathbf{D}_j^k))^T \phi^k(\mathbf{D}_j^k) \mathbf{w}_j^k \right] \\ &= \arg \min_j \sum_{k=1}^K \left[ -2 \text{Ker}(\mathbf{D}_j^k, \mathbf{y}^k) \mathbf{w}_j^k \right. \\ &\quad \left. + (\mathbf{w}_j^k)^T \text{Ker}(\mathbf{D}_j^k, \mathbf{D}_j^k) \mathbf{w}_j^k \right], \\ &\quad j = 1, 2, \dots, J. \end{aligned} \quad (19)$$

Based upon the above analysis, we can summarize the procedure of the proposed cMTL in the following Algorithm 1. The dominant computational cost of Algorithm 1 comes from the generalized gradient mapping step, whose computational demand is  $o(\sum_{j=1}^J \sum_{k=1}^K [\hat{K}_j^k + (\hat{K}_j^k + p_j b \tilde{I})])$ , where  $\hat{K}_j^k$  and  $\tilde{K}_j^k$  refer to the costs of kernels  $\text{Ker}(\mathbf{D}_j^k, \mathbf{y}^k)$  and  $\text{Ker}(\mathbf{D}_j^k, \mathbf{D}_j^k)$ , respectively. It is notable that different kernels involve different  $\hat{K}_j^k$  and  $\tilde{K}_j^k$  values. For instance, when the kernels degenerate into the linear scenario, the computational cost yields  $o(K \sum_{j=1}^J [bp_j + 2p_j b \tilde{I}])$ .

---

#### Algorithm 1 The proposed cMTL

---

**Require:** The training feature matrix  $\mathbf{D}^k$ ,  $k = 1, 2, \dots, K$ , the test feature matrix  $\mathbf{y}^k$ ,  $k = 1, 2, \dots, K$ , the regularization parameter  $\lambda$ , the step size parameter  $\mu$ , the number of iterations  $\tilde{I}$ , and the kernel function  $\text{Ker}$

**Ensure:** The class label of the test sample  $\mathbf{y}$

- 1: **for all**  $j = 1, 2, \dots, J$  **do**
- 2:   **Initialization:**  $i = 0$ ,  $\alpha^{(0)} = 1$  and  $\mathbf{w}_j^{k,(0)} = \mathbf{v}_j^{k,(0)} = 0$
- 3:   **while**  $i \leq \tilde{I}$  **do**
- 4:     **The generalized gradient mapping step:**
- 5:      $\mathbf{w}_j^{k,(i+(1/2))} = \mathbf{v}_j^{k,(i)} - \mu[-\text{Ker}(\mathbf{D}_j^k, \mathbf{y}^k) + \text{Ker}(\mathbf{D}_j^k, \mathbf{D}_j^k) \mathbf{v}_j^{k,(i)}]$ ,  $k = 1, 2, \dots, K$
- 6:      $\mathbf{w}_j^{(i+1)} = [1 - (\lambda\mu/\|\mathbf{w}_j^{(i+(1/2))}\|_2)]_+ \mathbf{w}_j^{(i+(1/2))}$
- 7:     **The aggregation forward step:**
- 8:      $\alpha^{(i+1)} = (2/(i+3))$
- 9:      $\mathbf{v}_j^{k,(i+1)} = \mathbf{w}_j^{k,(i+1)} + (((1-\alpha^{(i)})\alpha^{(i+1)})/\alpha^{(i)}) (\mathbf{w}_j^{k,(i+1)} - \mathbf{w}_j^{k,(i)})$ ,  $k = 1, 2, \dots, K$
- 10:    Update  $i$  by  $i = i + 1$
- 11:   **end while**
- 12: **end for**

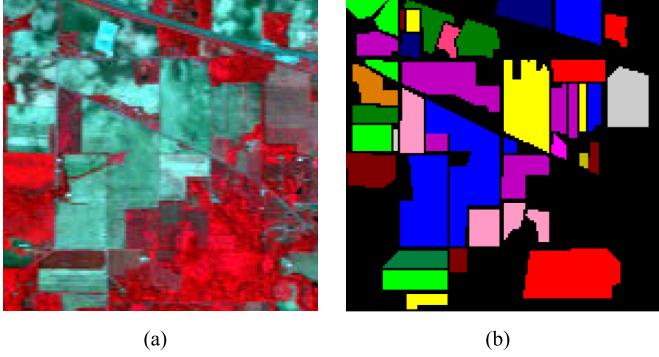


Fig. 4. Indian Pines data. (a) Three-band false-color composite. (b) Ground truth data with 16 classes.

### 13: The classification decision rule:

$$\begin{aligned} 14: \text{Calculate the class label of } \mathbf{y} \text{ by Class } \mathbf{y} &= \arg \min_j r_j = \\ &= \arg \min_j \sum_{k=1}^K [-2 \operatorname{Ker}(\mathbf{D}_j^k, \mathbf{y}^k) \mathbf{w}_j^k + (\mathbf{w}_j^k)^T \operatorname{Ker}(\mathbf{D}_j^k, \mathbf{D}_j^k) \mathbf{w}_j^k] \end{aligned}$$

## IV. EXPERIMENTS

In this section, we assess the effectiveness of the proposed method (termed as fTEMD-cMTL) with three benchmark HSI data cubes. The experimental results are compared visually and quantitatively with those obtained from several state-of-the-art methods, including 1D-EMD [21], 2D-EMD [29], and 3D-WT [33] for generating features and the classical SVM [39], the SVM with composite kernel (SVMCK) [49], the JSM solved by kernelized simultaneous orthogonal matching pursuit (KSOMP) [42], kernelized class-dependent SRC (KcdSRC) [45], spatial-aware dictionary learning (SADL)-based classifier [60], NRJSR [56], and the original MTL [21] for classification.

### A. Hyperspectral Data Sets

Three real-world HSI data sets, namely, Indian Pines data, University of Pavia data, and Salinas data, with available ground truth and different spectral and spatial resolutions are used for experiments and comparisons.

- 1) *Indian Pines data*: The first data set was collected by the Airborne Visible/Infrared Imaging Spectrometer (AVIRIS) sensor over a mixed forest/agricultural site from Northwestern Indiana, USA, on June 12, 1992. The original data set contains 220 spectral bands covering the 0.4–2.5- $\mu\text{m}$  range. In the experiments, the number of bands is reduced to 200 by removing 20 water vapor absorption bands. The spatial resolution is 20 m per pixel, and the spatial size is 145  $\times$  145. Fig. 4 depicts the three-band false-color composite image and the corresponding ground truth. This data set contains 16 classes of interest and a total of 10 366 labeled samples ranging unbalanced from 20 to 2468 for different classes, which poses a big challenge for the classification task. The detailed number of samples for each class is listed in Table I, whose background color demonstrates different classes of land covers.

- 2) *University of Pavia data*: The second data set was acquired by the Reflective Optics System Imaging Spectrometer (ROSIS) optical sensor over an urban area surrounding the University of Pavia in northern Italy on July 8, 2002. The original data set contains 115 spectral bands covering the 0.43–0.86- $\mu\text{m}$  range. Twelve noisy channels were removed, and 103 bands remained for experiments. This data set has a size of 610  $\times$  340 pixels with a spatial resolution of 1.3 m per pixel. Fig. 5 displays the false-color composite image, the ground truth data, and the available training samples. Table I shows the number of samples for each class and the available training samples. Similarly, the background color also corresponds to different classes of land covers. As listed in Table I, this data set contains nine classes of interest, and each class has more than 900 samples. However, the available training samples of each class are less than 600.
- 3) *Salinas data*: The third data set was captured by the AVIRIS sensor over Salinas Valley, California, USA, on October 8, 1998. The original data set consists of 224 spectral channels; 20 water absorption bands are discarded, and 204 bands are preserved in the experiments. The spatial size is 512  $\times$  217, and the spatial resolution is 3.7 m per pixel. Fig. 6 plots the false-color image and the corresponding ground truth. This data set consists of 16 classes of land covers and 54 129 labeled samples. The specific classes and the number of samples are shown in Table I, whose background color also distinguishes various classes.

### B. Experimental Design

Two experiments are designed to evaluate the effectiveness of the proposed fTEMD-cMTL. 1) We compare the fTEMD-based features with the raw spectral profiles (denoted as “Spec”) and 1D-EMD-, 2D-EMD-, and 3D-WT-based features. SVM and cMTL are adopted as classifiers in this experiment. 2) We compare the cMTL with several state-of-the-art classifiers, including SVM, SVMCK, KSOMP, and MTL. fTEMD is used for feature extraction in this experiment. All of those methods are compared numerically [overall accuracy (OA) and average accuracy (AA)] and statistically [kappa coefficient ( $\kappa$ )]. All of the experiments are implemented with MATLAB on a platform with Intel(R) Xeon(R) CPU (3.3 GHz), 8-GB RAM, and Windows 7 operating system. The experimental results are the average accuracy levels from ten times of repeated random sample selection. It should be pointed out here that the class labels of samples are difficult and expensive to be identified in reality. To validate the effectiveness of the proposed method in a limited-training-sample scenario, only ten samples from each class in the Indian Pines data and Salinas data are randomly selected as training samples, and the rest of the data are taken as test samples. Since the available training samples are separate from the whole samples in the University of Pavia data, only ten samples from each class of the available training samples are randomly chosen for training.

Moreover, several parameters need to be tuned in the experiments. In experiment 1, the stopping criterion of 1D-EMD is

TABLE I  
NUMBER OF SAMPLES (NoS) AND AVAILABLE TRAINING SAMPLES (NoATS) USED IN THE EXPERIMENTS

Indian Pines Data			University of Pavia Data			Salinas data			
Class	Name	NoS	Class	Name	NoS	NoATS	Class	Name	NoS
1	alfalfa	54	1	asphalt	6631	548	1	brocoli-green-weeds-1	2009
2	corn-no till	1434	2	meadows	18649	540	2	brocoli-green-weeds-2	3726
3	corn-min till	834	3	gravel	2099	392	3	fallow	1976
4	corn	234	4	trees	3064	524	4	fallow-rough-plow	1394
5	grass/pasture	497	5	metal sheets	1345	265	5	fallow-smooth	2678
6	grass/trees	747	6	bare soil	5029	532	6	stubble	3959
7	grass/pasture-mowed	26	7	bitumen	1330	375	7	celery	3579
8	hay-windrowed	489	8	bricks	3682	514	8	grapes-untrained	11271
9	oats	20	9	shadows	947	231	9	soil-vinyard-develop	6203
10	soybean-no till	968					10	corn-senesced-green-weeds	3278
11	soybean-min till	2468					11	lettuce-romaine-4wk	1068
12	soybean-clean till	614					12	lettuce-romaine-5wk	1927
13	wheat	212					13	lettuce-romaine-6wk	916
14	woods	1294					14	lettuce-romaine-7wk	1070
15	bldg-grass-tree-drives	380					15	vinyard-untrained	7268
16	stone-steel towers	95					16	vinyard-vertical-trellis	1807
Total		10366	Total		42776	3921	Total		54129

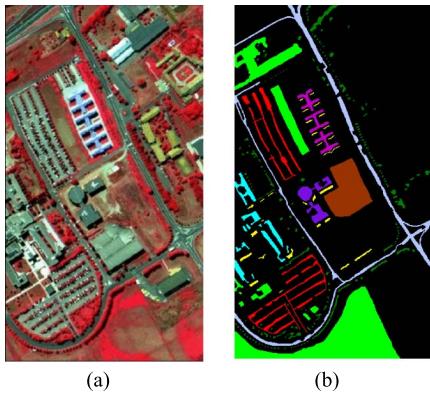


Fig. 5. University of Pavia data. (a) Three-band false-color composite. (b) Ground truth data with nine classes. (c) Available training samples.

set the same as that in [21], 2D-EMD is accelerated by the order statistics filters [69], 3D-WT is implemented for two levels with the “Haar” mother wavelet, and the subcubes are taken as spectral–spatial features of the HSI data sets. Based on the procedure of fTEMD (see Section II-A), we complete the whole decomposition until the residue  $R_{i+1}(m, n, b)$  generated in the  $(i + 1)$ th iteration contains less than three extrema, and therefore, the number of TIMFs can be automatically obtained. For this reason, five (or six and five) TIMFs are gained in the Indian Pines data (or University of Pavia data and Salinas data) by virtue of fTEMD. The components (i.e., IMFs/TIMFs or wavelet features) obtained by 1D-EMD, 2D-EMD, 3D-WT, and fTEMD can be treated, respectively, as different tasks in the cMTL, whereas the “Spec” can also be classified by the cMTL with only one task (i.e.,  $K = 1$ ). In experiment 2, the widely used radial basis function (RBF) kernel is adopted in almost all of the classifiers (i.e., SVM, SVMCK, KSOMP, KcdSRC, MTL, and cMTL) except the SADL and NRJSR, and the RBF parameter  $\gamma$  is tuned with  $\gamma \in \{2^{-8}, 2^{-7}, \dots, 2^8\}$ . According to [56] and [60], a linear kernel is used in SADL and NRJSR. For convenience, the penalty term in SVM, SVMCK, and SADL is set to 60. Note that each task is equally treated in the cMTL; to be fair, the composite weights of SVMCK are set to  $1/K$  for all of the TIMFs. As to the KSOMP, KcdSRC, and NRJSR, the sparsity level ranges from 10 to

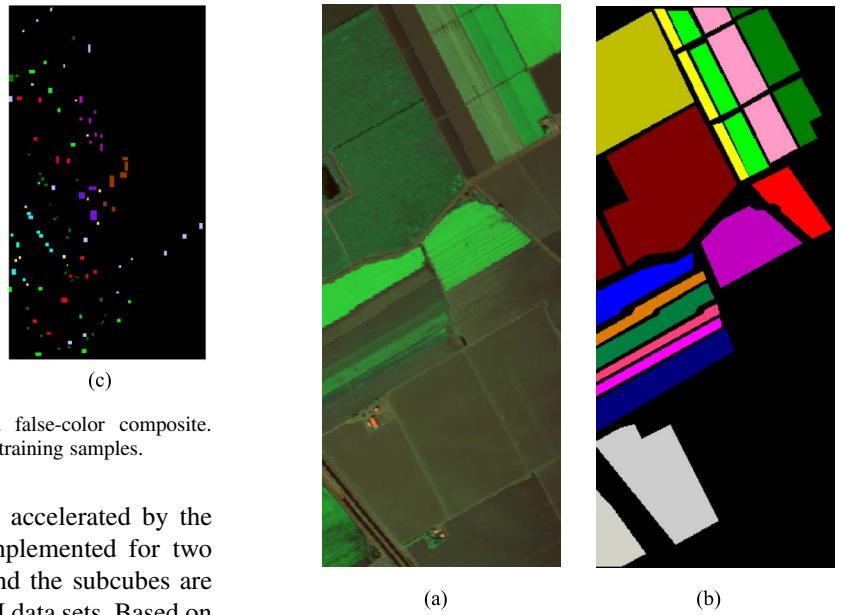


Fig. 6. Salinas data. (a) Three-band false-color composite. (b) Ground truth data with 16 classes.

100, and the spatial window sizes of KSOMP and NRJSR are chosen as  $9 \times 9$ ,  $5 \times 5$ , and  $9 \times 9$  in the Indian Pines data, University of Pavia data, and Salinas data, respectively. The regularization parameters of KcdSRC, SADL, and NRJSR are set the same as in [45], [56], and [60], respectively. For the MTL and cMTL, the regularization parameter  $\lambda$  is tuned in the range of  $\{0.01, 0.1, \dots, 1\}$ .

### C. Classification Results of Experiment 1

In this section, we provide the results of experiment 1 to validate the performance of the proposed fTEMD in extracting the spectral–spatial features of HSI. To this end, the fTEMD-based features are compared with the “Spec”-, 1D-EMD-, 2D-EMD-, and 3D-WT-based features, all of which are classified by the SVM and cMTL. The “Spec” and 1D-EMD exploit spectral features, 2D-EMD generates spatial features, whereas 3D-WT and fTEMD treat the HSI as a 3-D cube and obtain spectral–spatial features. As an example, Fig. 7 shows the

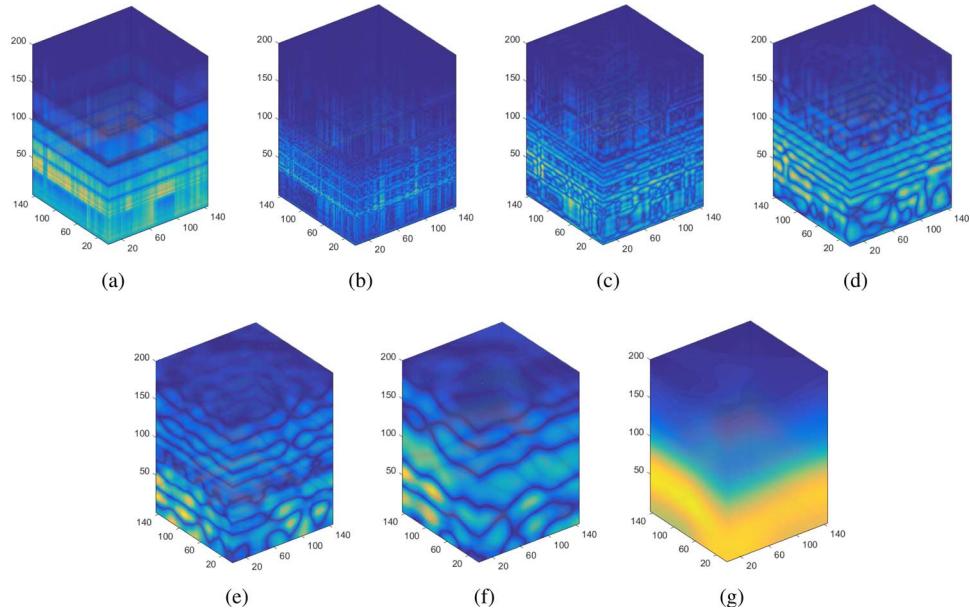


Fig. 7. Decomposition results of fTEMD in the Indian Pines data. (a) Original data cube. (b)–(f) First to fifth TIMFs. (g) Final residue.

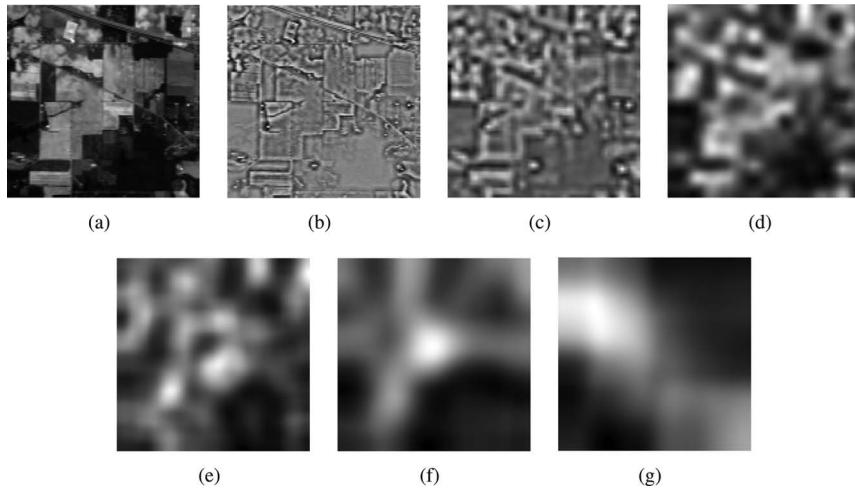


Fig. 8. Decomposition results of the 30th band by fTEMD in the Indian Pines data. (a) The 30th band from the original data cube. (b)–(f) The 30th band from first to fifth TIMFs. (g) The 30th band from the final residue.

decomposition results of fTEMD in the Indian Pines data, and Fig. 8 plots the decomposition results of the 30th band by fTEMD. In general, the lower-order TIMFs contain higher local spatial frequencies and, thus, have less spatial integration. On the contrary, the higher-order TIMFs involve lower local spatial frequencies and more spatial integration. Moreover, we also depict the “Spec” and fTEMD features of classes 4, 7, 9, and 15<sup>3</sup> of the test samples in the Indian Pines data, under the 2-D condition. Fig. 9(a) shows the original spectral features of band #40 and #70, whereas Fig. 9(b) gives the fifth fTIMF features of band #40 and #70. Comparing Fig. 9(a) and (b), the class separability is greatly improved by the fTEMD. Fig. 10 plots the normalized residuals of a sample from class 9 located at (70,23) in the Indian Pines data. The fTEMD-based features yield more accurate classification results than “Spec” regardless

<sup>3</sup>Classes 4, 7, 9, and 15 are chosen to form the scatterplots because those classes of land covers are difficult to separate.

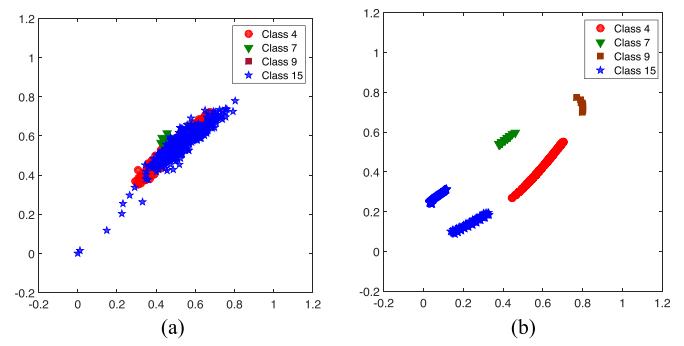


Fig. 9. Two-dimensional representation of the “Spec” and fTEMD features of the test samples in the Indian Pines data. (a) “Spec” features. (b) fTEMD features.

of the classifiers employed. Therefore, it is more possible to gain better classification performance with the fTEMD features. The classification maps of different methods are depicted

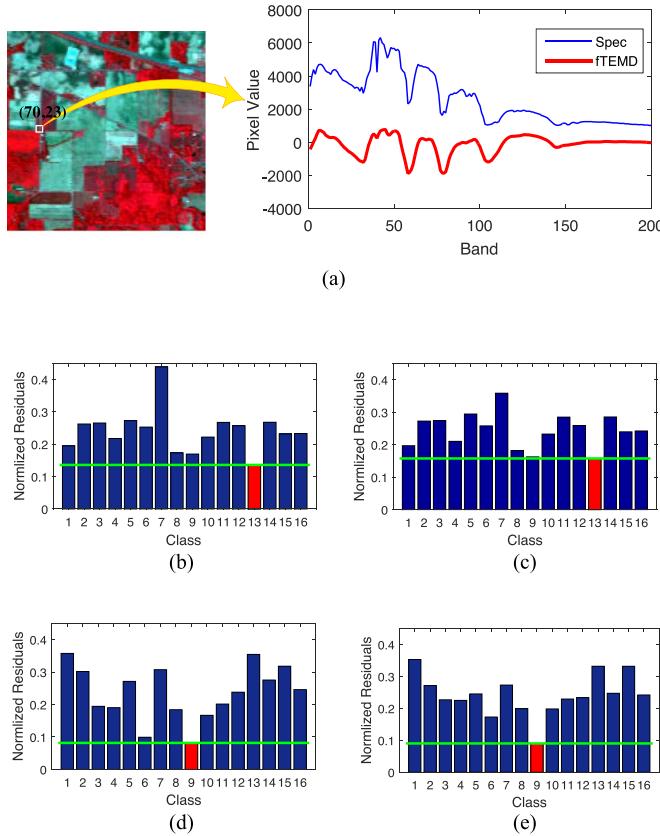


Fig. 10. Normalized residuals of a sample from class 9 located at (70,23). (a) Detailed location, the original (“Spec”) and reconstructed (fTEMD) signatures of the sample in class 9. (b) and (c) Normalized residuals of MTL and cMTL with “Spec” features, respectively. (d) and (e) Normalized residuals of MTL and cMTL with fTEMD features, respectively.

in Figs. 11–13. One can roughly see that the fTEMD-based features yield less classification errors than other methods, which demonstrates that the fTEMD can significantly improve the classification performance. Since the fTEMD is based on FATEMD and the major improvement of fTEMD is the reduction of calculation burden, we compare the computational time of FATEMD and fTEMD in Table II. Moreover, Tables III–V compare the classification accuracy of each class, OA, AA,  $\kappa$ , and computational time of various methods. A few observations can be made from Figs. 11–13 and Tables II–V.

- 1) “Spec” and 1D-EMD generate more “salt-and-pepper-like” appearance than other methods. As can be observed from Figs. 11–13, there exist many scattered salt-and-pepper-like misclassified samples with “Spec”- and 1D-EMD-based features, whereas the classification errors of 2D-EMD, 3D-WT, and fTEMD are spatially concentrated. Moreover, Tables III–V quantitatively compare the performance of various methods. As shown in Tables III–V, the spectral-characteristics-based methods (i.e., “Spec” and 1D-EMD) provide poorer classification accuracy levels compared with other methods that take into consideration the spatial information. For instance, the OA of “Spec” in Table III is about 10%–20% lower than 2D-EMD, 3D-WT, and fTEMD. This is because the HSI data provide abundant information in spatial do-

mains, and spatial information is a valuable complement to spectral information. As a result, the aforementioned observations validate the importance of spatial information in hyperspectral classification.

- 2) We then compare different feature extraction methods using the same classifiers. It is notable from Tables III–V that the classification accuracy of “Spec” is not very satisfactory since all of the OAs, AAs, and  $\kappa$  coefficients are the lowest among all feature extraction methods. By contrast, 1D-EMD, 2D-EMD, and 3D-WT perform much better than “Spec.” For instance, the OA of “Spec” is lower than 70% with an SVM classifier in Table III, whereas the OAs of 1D-EMD, 2D-EMD, and 3D-WT are much higher than 70% with the same classifier. This is due to the fact that 1D-EMD, 2D-EMD, and 3D-WT are able to extract significant features from the original HSI data sets. Moreover, the classification performance is further increased with the proposed approach. As shown in Table III, the fTEMD-based features increase the OA compared with the 3D-WT method by about 6%–7%. The classification accuracy levels of the difficult-to-separate classes (e.g., classes 4, 7, 9, and 15) are also improved by fTEMD, which is consistent with the conclusions drawn from Fig. 9. In addition, as depicted in Fig. 11, the classification maps of fTEMD are closer to the ground truth [see Fig. 4(b)] than other methods. As in the above analysis, features play a vital role in hyperspectral classification, and fTEMD brings significant advantages to spectral–spatial feature extraction.
- 3) As to the effectiveness of classifiers, the cMTL performs much better than SVM with the same feature extraction methods. For instance, as noted in Table IV, an improvement of more than 5% will be gained by the cMTL if the same features are involved. One can also observe from Fig. 12 that the classification errors of SVM are significantly reduced by the cMTL. Similar classification performance can be observed from Figs. 11 and 13 and Tables III and V that the proposed cMTL achieves better performance than SVM in visual interpretation and accuracy. Therefore, the aforementioned experimental results demonstrate the effectiveness of the proposed cMTL method.
- 4) For the computational cost issue, SVM is faster than cMTL, whereas fTEMD spends moderate computational time among the feature extraction methods. On the one hand, SVM is implemented by LibSVM,<sup>4</sup> which adopts C++ software and MATLAB interface to accelerate the process. Note that the implementation of cMTL utilizes MATLAB only; cMTL takes more time cost than SVM. For instance, with “Spec”-based features, the runtime of cMTL is about 209.74 s, which is nearly 200 s longer than that of the SVM in Table V. Similar phenomena can also be found from the Indian Pines data and University of Pavia data. On the other hand, the computational time of fTEMD is much less than that of 1D-EMD and

<sup>4</sup><http://www.csie.ntu.edu.tw/~cjlin/index.html>

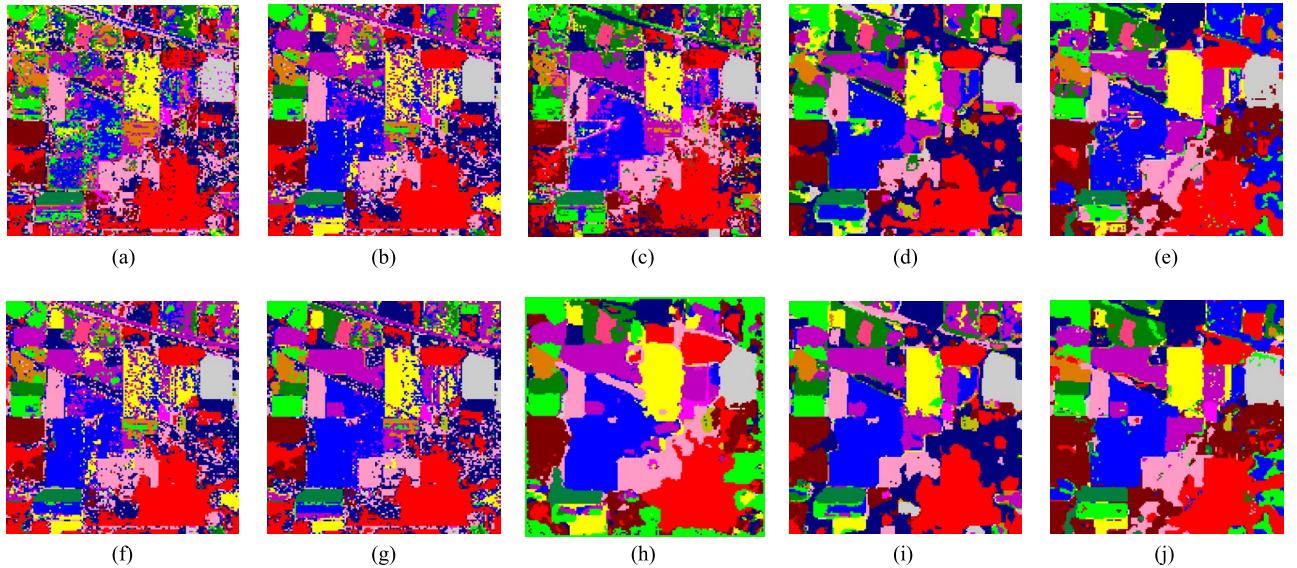


Fig. 11. Classification maps of the Indian Pines data with (a) Spec(SVM)-, (b) 1D-EMD(SVM)-, (c) 2D-EMD(SVM)-, (d) 3D-WT(SVM)-, (e) fTEMD(SVM)-, (f) Spec(cMTL)-, (g) 1D-EMD(cMTL)-, (h) 2D-EMD(cMTL)-, (i) 3D-WT(cMTL)-, and (j) fTEMD(cMTL)-based methods.

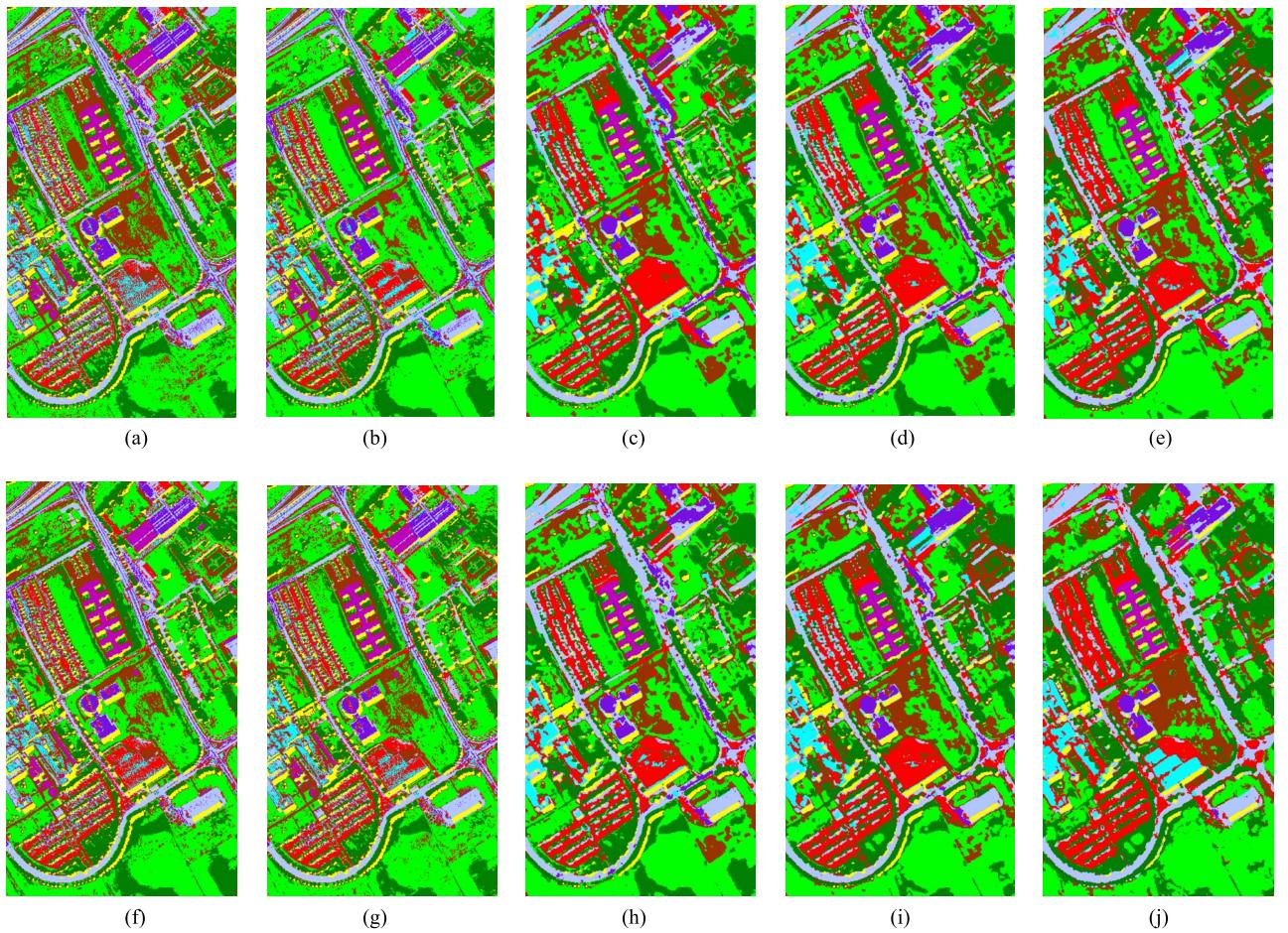


Fig. 12. Classification maps of the University of Pavia data with (a) Spec(SVM)-, (b) 1D-EMD(SVM)-, (c) 2D-EMD(SVM)-, (d) 3D-WT(SVM)-, (e) fTEMD(SVM)-, (f) Spec(cMTL)-, (g) 1D-EMD(cMTL)-, (h) 2D-EMD(cMTL)-, (i) 3D-WT(cMTL)-, and (j) fTEMD(cMTL)-based methods.

2D-EMD but longer than that of 3D-WT, whereas “Spec” takes 0 s since no feature extraction method is performed. As shown in Table V, fTEMD takes 570.54 s, which is

about 2600 and 5600 s faster than 1D-EMD and 2D-EMD, respectively. This is due to the fact that 3D-DT and separable filters are used to speed up the fTEMD.

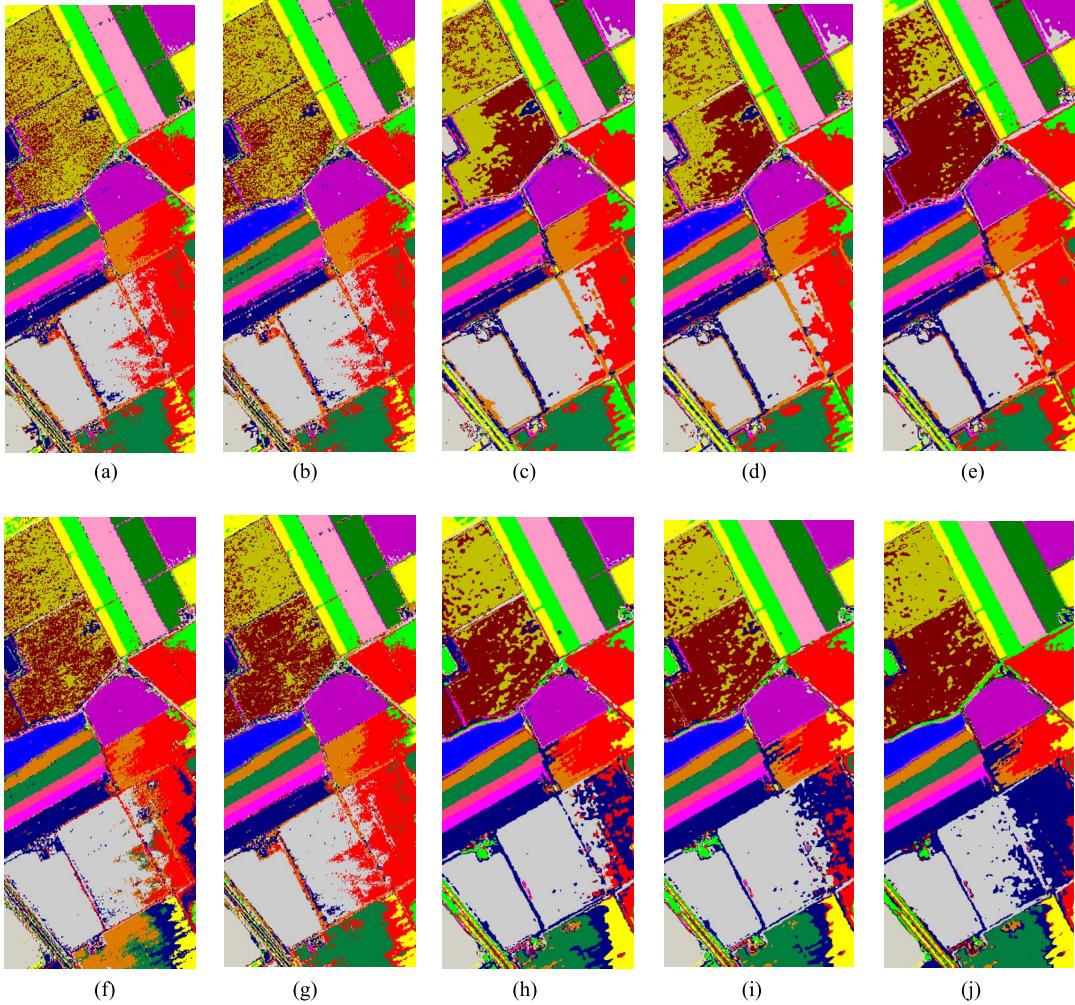


Fig. 13. Classification maps of the Salinas data with (a) Spec(SVM)-, (b) 1D-EMD(SVM)-, (c) 2D-EMD(SVM)-, (d) 3D-WT(SVM)-, (e) fTEMD(SVM)-, (f) Spec(cMTL)-, (g) 1D-EMD(cMTL)-, (h) 2D-EMD(cMTL)-, (i) 3D-WT(cMTL)-, and (j) fTEMD(cMTL)-based methods.

TABLE II  
COMPUTATIONAL TIME (S) OF FATEMD AND FTEMD  
FOR DIFFERENT DATA SETS

Data Sets	FATEMD	fTEMD
Indian Pines data	24268.37	240.25
University of Pavia data	55965.81	540.47
Salinas data	59406.25	570.54

The advantage of fTEMD will become more obvious if larger sizes of data sets are involved. Moreover, as shown in Table II, fTEMD is also much faster than the existing FATEMD. For instance, FATEMD consumes more than 6 h in dealing with the Indian Pines data with the size of  $145 \times 145 \times 200$ , whereas only about 4 min is required in fTEMD. As such, the proposed fTEMD-cMTL is satisfactory in terms of computational efficiency compared with other methods.

#### D. Classification Results of Experiment 2

In experiment 2, the proposed cMTL is compared with some widely used classifiers including the SVM, SVMCK, KSOMP, KcdSRC, SADL, NRJSR, and MTL. In SVM, KSOMP,

KcdSRC, SADL, and NRJSR, the sum of TIMFs<sup>5</sup> are taken as features, whereas each TIMF is treated as a kind of feature in SVMCK, MTL, and cMTL. Figs. 14–16 depict the classification maps of different methods for the three experimental HSI data sets. It is observed that the SVM has the most classification errors, whereas the results of cMTL are closer to the ground truth [see Figs. 4(b), 5(b), and 6(b)] than other methods. The detailed classification results of various methods are shown in Tables VI–VIII, which display the classification accuracy of each class, OA, AA,  $\kappa$ , and computational time. Since SVM and cMTL are used as classifiers in experiment 1 and fTEMD-based features are adopted in experiment 2, the classification results of SVM and cMTL in experiment 2 are the same as those of fTEMD(SVM) and fTEMD(cMTL) in experiment 1, respectively. Three main results can be highlighted from Figs. 14–16 and Tables VI–VIII.

- 1) Although SVM is one of the state-of-the-art methods in hyperspectral classification, its classification accuracy

<sup>5</sup>Although the sums contain multiple cases (e.g., TIMF1+TIMF2, TIMF1+TIMF2+TIMF3, etc.), we choose only one case that leads to the highest classification accuracy.

TABLE III

CLASSIFICATION ACCURACY (%) AND COMPUTATIONAL TIME OF DIFFERENT METHODS FOR THE INDIAN PINES DATA IN EXPERIMENT 1

Class	SVM					cMTL				
	Spec	1D-EMD	2D-EMD	3D-WT	fITEMD	Spec	1D-EMD	2D-EMD	3D-WT	fITEMD
1	86.36	88.64	88.64	90.91	95.46	88.64	90.91	93.18	95.46	97.73
2	39.75	64.64	70.93	77.11	82.16	61.87	83.18	83.43	94.21	95.26
3	54.85	56.25	63.35	60.92	87.50	55.34	68.69	74.27	83.25	97.39
4	76.34	78.80	75.45	75.67	72.99	78.57	79.02	78.13	82.81	87.05
5	85.83	83.78	81.11	85.22	89.84	75.77	76.69	85.63	83.27	94.15
6	84.80	92.54	83.58	90.50	83.04	90.91	93.28	93.89	95.66	90.10
7	56.25	68.75	68.75	68.75	75.00	68.75	68.75	68.75	71.88	81.25
8	90.19	98.33	98.75	98.02	93.84	98.75	99.58	98.54	99.58	95.51
9	30.00	60.00	70.00	75.00	85.00	60.00	70.00	70.00	80.00	100.00
10	60.33	63.94	71.29	69.62	89.88	66.70	85.07	92.59	89.09	93.74
11	46.34	70.00	71.52	71.77	86.05	70.99	77.77	80.31	79.13	95.57
12	37.25	38.99	55.80	62.67	73.01	38.25	60.43	74.50	82.29	81.79
13	98.52	98.02	99.01	99.01	98.52	99.51	99.26	98.02	99.51	98.02
14	84.50	96.85	88.08	95.79	85.28	96.81	96.03	97.04	98.17	96.89
15	52.97	54.19	66.49	83.65	90.54	54.32	66.89	78.11	84.46	98.24
16	83.53	93.53	90.59	94.12	100.00	95.29	96.47	98.82	97.65	100.00
OA	60.82	73.17	75.19	78.47	85.70	72.72	82.09	85.67	88.33	94.37
AA	66.74	75.45	77.71	81.17	86.76	75.03	82.00	85.33	88.53	93.92
$\kappa$	56.20	69.39	71.92	75.54	83.79	68.87	79.58	83.70	86.71	93.57
time(s) <sup>1</sup>	0 +1.93	1152.25 +1.89	2580.37 +1.85	8.12 +2.17	240.25 +1.91	0 +35.50	1152.25 +35.45	2580.37 +34.58	8.12 +36.41	240.25 +35.35

<sup>1</sup> The first and second rows denote the computational time of feature extraction methods and classification methods, respectively.

TABLE IV

CLASSIFICATION ACCURACY (%) AND COMPUTATIONAL TIME OF DIFFERENT METHODS FOR THE UNIVERSITY OF PAVIA DATA IN EXPERIMENT 1

Class	SVM					cMTL				
	Spec	1D-EMD	2D-EMD	3D-WT	fITEMD	Spec	1D-EMD	2D-EMD	3D-WT	fITEMD
1	65.83	61.55	63.77	66.29	68.99	64.18	66.48	73.81	80.49	84.39
2	61.58	76.88	77.66	78.59	85.66	68.84	79.25	79.41	84.57	85.82
3	59.01	60.11	59.94	63.31	72.34	84.79	79.56	79.78	75.37	85.45
4	94.99	97.05	94.85	96.50	94.16	96.81	96.63	96.39	96.39	99.14
5	99.28	99.28	99.64	99.73	99.64	99.28	99.37	99.64	99.46	99.82
6	56.67	56.71	57.11	58.29	67.32	57.66	63.06	63.60	66.62	82.52
7	86.65	87.36	89.30	86.95	88.18	84.51	84.81	85.63	88.28	96.23
8	64.30	72.59	87.49	91.91	94.08	76.31	82.19	91.68	93.22	96.58
9	98.62	99.37	99.25	99.37	99.12	98.99	99.25	99.87	99.50	99.75
OA	66.63	73.83	75.72	77.27	82.38	72.05	78.01	80.13	83.85	88.00
AA	76.32	78.99	81.00	82.33	85.50	81.26	83.40	85.54	87.10	92.19
$\kappa$	58.22	65.95	68.24	70.13	76.60	64.26	71.31	74.00	78.60	84.21
time(s) <sup>1</sup>	0 +2.31	2993.45 +2.21	5922.64 +2.15	20.87 +2.35	540.47 +2.26	0 +154.97	2993.45 +154.47	5922.64 +153.77	20.87 +155.04	540.47 +154.07

<sup>1</sup> The first and second rows denote the computational time of feature extraction methods and classification methods, respectively.

TABLE V

CLASSIFICATION ACCURACY (%) AND COMPUTATIONAL TIME OF DIFFERENT METHODS FOR THE SALINAS DATA IN EXPERIMENT 1

Class	SVM					cMTL				
	Spec	1D-EMD	2D-EMD	3D-WT	fITEMD	Spec	1D-EMD	2D-EMD	3D-WT	fITEMD
1	97.30	96.95	97.05	97.05	98.70	98.90	99.70	99.50	99.85	100.00
2	98.52	98.84	98.57	98.52	98.68	99.19	99.14	99.53	99.57	99.80
3	98.63	99.44	96.08	97.41	98.02	99.75	99.80	99.39	99.85	99.97
4	99.35	99.35	98.63	98.63	98.63	98.99	99.49	99.46	99.42	99.64
5	97.49	96.85	91.53	91.04	95.05	97.71	97.75	97.81	98.58	99.64
6	99.75	99.75	99.97	99.97	99.97	99.90	99.85	99.28	100.00	99.99
7	99.33	99.33	96.53	96.58	96.39	99.64	99.52	99.69	99.64	99.86
8	35.81	43.58	52.70	54.89	73.21	64.89	74.55	80.49	81.32	90.09
9	95.61	95.59	92.22	92.18	95.12	98.85	99.90	98.61	99.10	99.64
10	80.11	81.88	78.00	80.84	80.81	92.41	93.70	94.77	95.78	98.93
11	91.87	91.87	90.17	90.45	93.57	97.92	100.00	98.82	99.81	99.95
12	99.43	95.93	99.74	99.58	99.74	99.11	99.90	99.79	99.95	100.00
13	97.90	97.79	99.01	99.01	99.56	99.45	99.67	99.67	99.78	99.78
14	93.30	93.58	91.04	91.13	93.68	96.60	99.43	98.58	99.72	100.00
15	74.30	74.48	75.57	78.31	70.45	74.32	79.10	77.19	83.08	88.81
16	89.04	91.04	94.60	97.16	97.50	98.83	99.72	98.61	99.94	99.67
OA	80.27	81.98	83.01	84.12	87.63	88.17	91.21	92.01	93.30	96.25
AA	90.48	91.02	90.71	91.42	93.07	94.78	96.33	96.32	97.21	98.49
$\kappa$	78.25	80.10	81.22	82.44	86.32	86.88	90.24	91.11	92.56	95.83
time(s) <sup>1</sup>	0 +2.48	3192.61 +2.39	6230.41 +2.36	22.81 +2.54	570.54 +2.41	0 +209.74	3192.61 +207.98	6230.41 +207.87	22.81 +209.76	570.54 +208.56

<sup>1</sup> The first and second rows denote the computational time of feature extraction methods and classification methods, respectively.

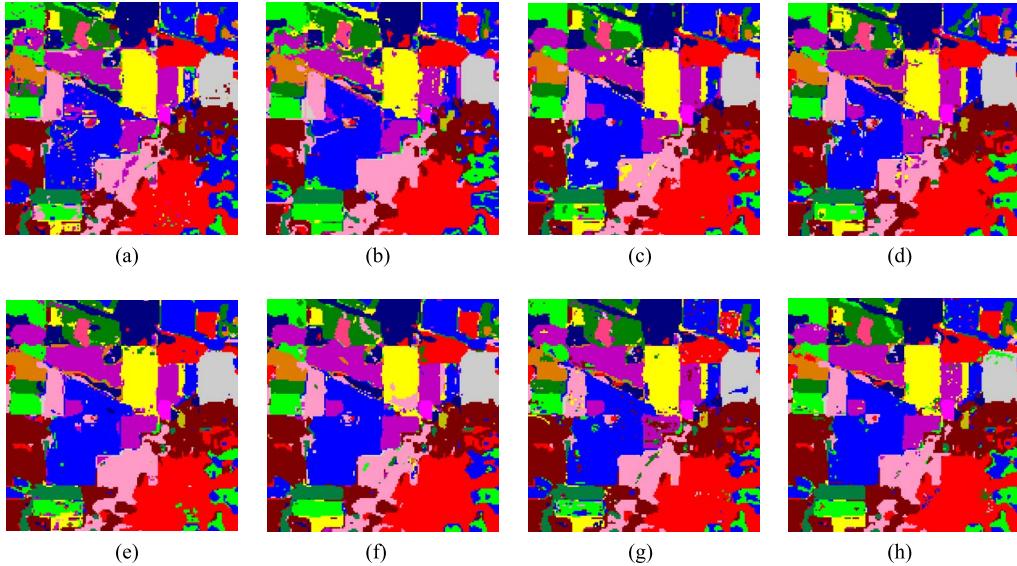


Fig. 14. Classification maps of the Indian Pines data classified by (a) SVM, (b) SVMCK, (c) KSOMP, (d) KcdSRC, (e) SADL, (f) NRJSR, (g) MTL, and (h) cMTL with fTEMD features.

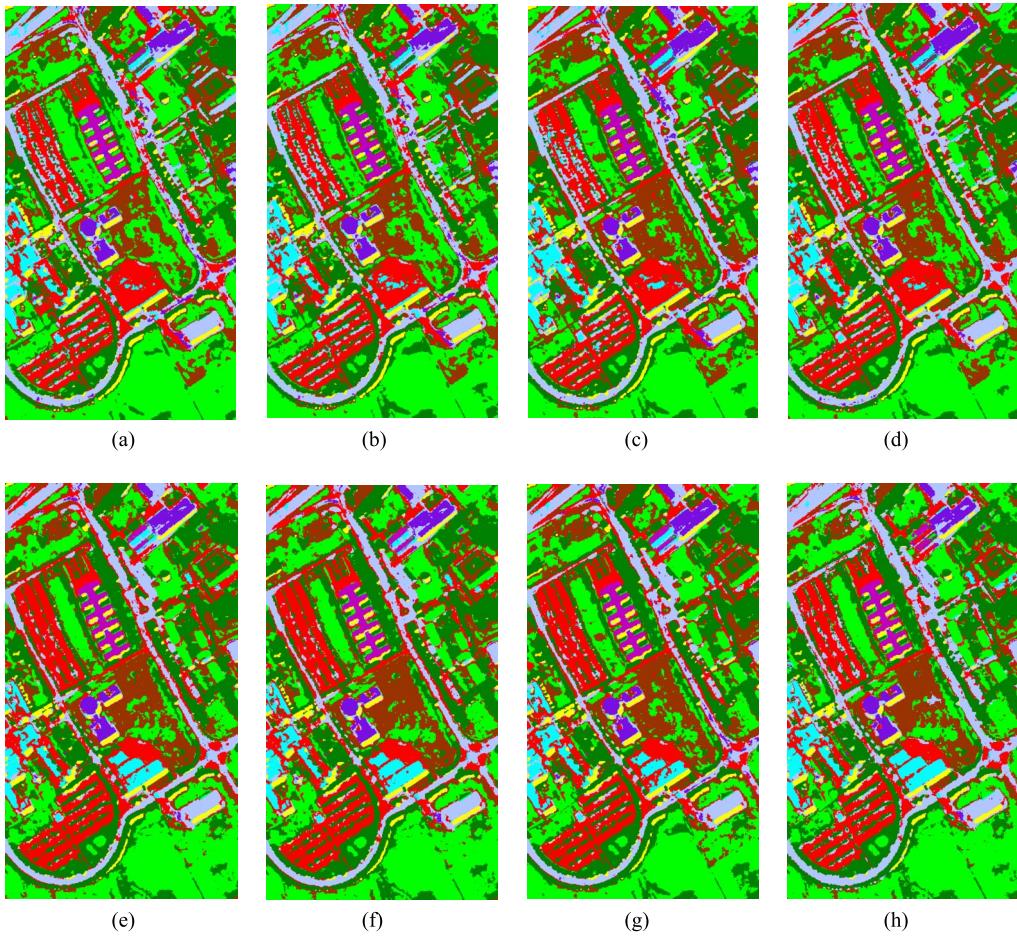


Fig. 15. Classification maps of the University of Pavia data classified by (a) SVM, (b) SVMCK, (c) KSOMP, (d) KcdSRC, (e) SADL, (f) NRJSR, (g) MTL, and (h) cMTL with fTEMD features.

is much lower than several recently proposed classifiers (e.g., SVMCK, KSOMP, KcdSRC, SADL, NRJSR, and MTL). For instance, as reported in Table VI, the OA of SVM is about 1%–4% lower than those of SVMCK,

KSOMP, KcdSRC, SADL, NRJSR, and MTL, respectively. Similar results can also be found in Tables VII and VIII. The reason behind this is that the SVMCK, KSOMP, KcdSRC, SADL, NRJSR, and MTL provide more

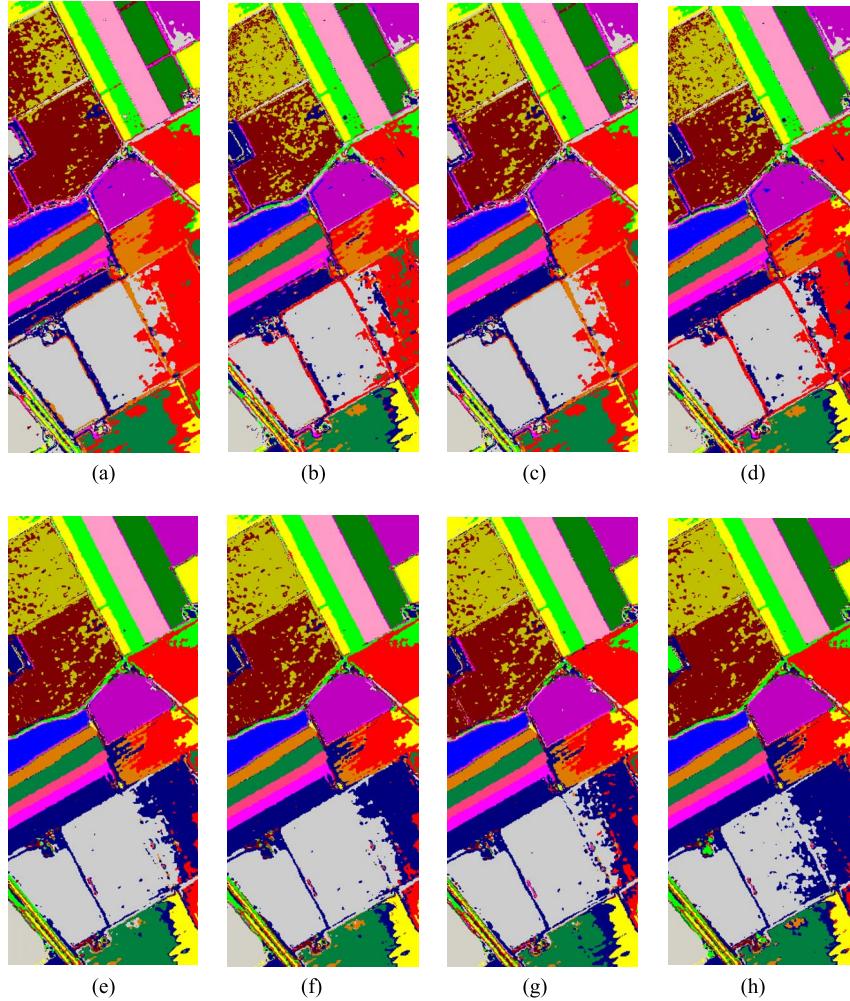


Fig. 16. Classification maps of the Salinas data classified by (a) SVM, (b) SVMCK, (c) KSOMP, (d) KcdSRC, (e) SADL, (f) NRJSR, (g) MTL, and (h) cMTL with fTEMd features.

superior performance in handling the spectral–spatial features of the HSI data sets than SVM. Moreover, as to SVMCK, KSOMP, KcdSRC, SADL, NRJSR, and MTL, NRJSR yields the highest classification accuracy; SADL and MTL are comparable; KSOMP and KcdSRC trail slightly behind NRJSR, SADL, and MTL; and SVMCK leads to lower OA, AA, and  $\kappa$  values than the other five methods.

2) We now analyze the performance of the proposed cMTL in detail. Compared with other widely used classifiers (i.e., SVM, SVMCK, KSOMP, KcdSRC, SADL, NRJSR, and MTL), cMTL gives better classification performance regardless of the data sets employed. As displayed in Table VI, although the classification accuracy of cMTL for each class is not always higher than that of the other methods, the OA, AA, and  $\kappa$  values of cMTL significantly increase. For instance, the OA of cMTL in the Indian Pines data is 8.67%, 7.41%, 6.06%, 7.01%, 4.43%, 3.35%, and 4.76% higher than those of the SVM, SVMCK, KSOMP, KcdSRC, SADL, NRJSR, and MTL, respectively. Similar observations can be made from Tables VII and VIII. Specifically, Fig. 10 compares the normalized residuals of a sample from class 9 located

at (70,23) in the Indian Pines data. It is notable that the normalized residuals of cMTL achieve better discriminative property than MTL. cMTL announces improved classification results because the features of the HSI cubes are significantly used, and the class-oriented information of the TIMFs is fully exploited. This stresses yet again the importance of classifiers for hyperspectral classification.

3) Finally, concerning the time cost of various approaches, we can see that SVM is the fastest; that SVMCK, KSOMP, KcdSRC, SADL, MTL, and the proposed cMTL are slower than SVM; and that NRJSR is the slowest. As shown in Table VIII, SVM and SVMCK take about 2 and 20 s to classify the Salinas data, respectively, whereas the computational time of cMTL is moderate (i.e., about 200 s) among the competitive methods. Similar phenomena can also be observed from Tables VI and VII. As explained in experiment 1, C++ software is adopted to speed up the implementation process of SVM/SVMCK and, thus, makes the SVM/SVMCK faster than other methods. The runtime of cMTL is also acceptable since only hundreds of seconds are required. Moreover, the speed of cMTL implementing can be further improved with the rapid development of computer

**TABLE VI**  
CLASSIFICATION ACCURACY (%) AND COMPUTATIONAL TIME OF DIFFERENT METHODS FOR THE INDIAN PINES DATA IN EXPERIMENT 2

Class	SVM	SVMCK	KSOMP	KcdSRC	SADL	NRJSR	MTL	cMTL
1	95.46	93.18	93.18	93.18	95.46	95.46	95.46	97.73
2	82.16	82.51	89.61	87.78	90.24	94.45	84.97	95.26
3	87.50	87.86	83.01	87.74	83.13	91.26	88.05	97.39
4	72.99	76.79	75.89	76.34	88.39	86.61	89.06	87.05
5	89.84	90.97	91.79	88.09	89.32	90.35	86.24	94.15
6	83.04	83.85	78.97	77.21	85.89	80.46	79.38	90.10
7	75.00	75.00	81.25	75.00	81.25	75.00	75.00	81.25
8	93.84	94.57	100.00	98.33	99.17	99.37	94.05	95.51
9	85.00	90.00	0.00	80.00	90.00	50.00	95.00	100.00
10	89.88	90.81	93.63	91.34	85.80	82.05	88.47	93.74
11	86.05	86.33	88.65	85.40	92.64	94.79	93.41	95.57
12	73.01	73.68	72.35	70.03	78.64	79.64	80.63	81.79
13	98.52	97.53	97.03	97.03	97.03	96.04	98.02	98.02
14	85.28	91.12	92.60	95.25	92.52	94.16	96.22	96.89
15	90.54	91.62	91.35	92.43	97.57	93.24	92.70	98.24
16	100.00	100.00	100.00	98.82	100.00	100.00	100.00	100.00
OA	85.70	86.96	88.31	87.36	89.94	91.02	89.61	94.37
AA	86.76	87.86	83.08	87.12	90.44	87.68	89.79	93.92
$\kappa$	83.79	85.17	86.70	85.65	88.53	89.75	88.17	93.57
time(s) <sup>1</sup>	240.25	240.25	240.25	240.25	240.25	240.25	240.25	240.25
	+1.91	+9.51	+195.23	+251.84	+31.52	+1968.57	+47.98	+35.35

<sup>1</sup> The first and second rows denote the computational time of feature extraction methods and classification methods, respectively.

**TABLE VII**  
CLASSIFICATION ACCURACY (%) AND COMPUTATIONAL TIME OF DIFFERENT METHODS FOR THE UNIVERSITY OF PAVIA DATA IN EXPERIMENT 2

Class	SVM	SVMCK	KSOMP	KcdSRC	SADL	NRJSR	MTL	cMTL
1	68.99	71.24	86.25	82.98	79.55	80.38	75.22	84.39
2	85.66	85.16	83.29	81.87	84.67	84.78	85.37	85.82
3	72.34	72.51	74.16	72.95	72.95	75.65	74.10	85.45
4	94.16	96.43	98.08	98.15	98.52	98.76	98.04	99.14
5	99.64	99.73	100.00	98.83	99.91	99.82	99.82	99.82
6	67.32	69.55	77.62	75.07	80.16	86.26	78.61	82.52
7	88.18	88.28	87.56	91.95	94.80	94.39	88.79	96.23
8	94.08	93.97	94.98	95.87	95.78	96.02	95.27	96.58
9	99.12	99.25	46.79	99.12	98.99	62.26	99.62	99.75
OA	82.38	82.93	84.60	84.29	85.72	86.01	85.01	88.00
AA	85.50	86.23	83.19	88.53	89.48	86.48	88.32	92.19
$\kappa$	76.60	77.43	79.72	79.37	81.20	81.61	80.26	84.21
time(s) <sup>1</sup>	540.47	540.47	540.47	540.47	540.47	540.47	540.47	540.47
	+2.26	+19.87	+990.51	+ 1251.32	+83.56	+ 17312.51	+361.51	+154.07

<sup>1</sup> The first and second rows denote the computational time of feature extraction methods and classification methods, respectively.

**TABLE VIII**  
CLASSIFICATION ACCURACY (%) AND COMPUTATIONAL TIME OF DIFFERENT METHODS FOR THE SALINAS DATA IN EXPERIMENT 2

Class	SVM	SVM-CK	KSOMP	KcdSRC	SADL	NRJSR	MTL	cMTL
1	98.70	98.70	99.75	99.70	99.85	99.85	99.85	100.00
2	98.68	98.65	99.30	98.36	99.73	99.73	99.57	99.80
3	98.02	98.12	99.75	98.17	99.80	99.90	99.85	99.97
4	98.63	98.63	98.99	98.41	99.42	99.71	99.42	99.64
5	95.05	95.09	98.69	95.24	98.69	98.91	98.69	99.64
6	99.97	99.95	100.00	99.95	100.00	100.00	100.00	99.99
7	96.39	96.33	99.61	99.58	99.75	99.69	99.72	99.86
8	73.21	73.81	83.02	80.91	84.97	85.28	83.91	90.09
9	95.12	97.01	98.55	97.01	99.47	99.48	99.47	99.64
10	80.81	84.61	95.75	84.64	97.22	97.06	96.54	98.93
11	93.57	93.38	100.00	96.88	100.00	100.00	100.00	99.95
12	99.74	99.22	99.95	99.22	99.95	99.95	99.95	100.00
13	99.56	99.56	99.78	99.56	99.78	99.78	99.78	99.78
14	93.68	95.38	99.15	95.19	99.72	99.72	99.81	100.00
15	70.45	76.26	79.76	76.63	84.25	85.42	83.92	88.81
16	97.50	98.50	99.50	98.44	99.94	99.89	99.94	99.67
OA	87.63	89.03	93.09	90.86	94.37	94.61	94.06	96.25
AA	93.07	93.95	96.67	94.87	97.66	97.77	97.53	98.49
$\kappa$	86.32	87.82	92.32	89.84	93.74	94.00	93.40	95.83
time(s) <sup>1</sup>	570.54	570.54	570.54	570.54	570.54	570.54	570.54	570.54
	+2.41	+23.85	+1009.58	+1425.12	+ 101.57	+ 19841.72	+417.66	+208.56

<sup>1</sup> The first and second rows denote the computational time of feature extraction methods and classification methods, respectively.

hardware and software technology. In this regard, it can be declared that cMTL possesses a practical value in real applications.

In a nutshell, the classification results of experiments 1 and 2 on three benchmark HSI data sets validate the effectiveness of the proposed fTEMd-cMTL method in improving the classification performance.

## V. CONCLUSION

In this paper, a fTEMd-based feature extraction approach with cMTL classifier has been proposed for hyperspectral classification. fTEMd is proposed to naturally treat the HSI data sets as 3-D cubes and decompose the HSIs into several TIMFs. Note that extending the 1D/2D-EMD directly into the 3-D scenario requires complicated computation; two strategies are introduced to accelerate the TEMD. On the one hand, the distances among extrema, which can be applied to determine the filter sizes, are effectively calculated by the 3D-DT. On the other hand, separable filters are used to generate the envelopes in the shifting process. This means that a 1-D filter can be implemented three times to gain the same results of the original complicated 3-D filter. Moreover, to classify the fTEMd-based features, cMTL is subsequently proposed in this paper. The cMTL takes each TIMF as features of a particular task and obtains the representation coefficients by making full use of the class label information. The optimization problems of the cMTL are effectively solved by the APG, which can alternately update the weight matrix sequence and the aggregation matrix sequence. The class label of the test sample can be finally identified by the minimum reconstruction error. Experimental results on three benchmark HSI data sets have consistently confirmed the effectiveness of the fTEMd-cMTL. It is worth mentioning that the proposed fTEMd and cMTL methods can be widely used in the feature extraction of other 3-D data cubes (e.g., magnetic resonance imaging) and the simultaneous classification of multiple features. Therefore, future work could be devoted, for instance, to the applications of the proposed method for other classification cases.

## ACKNOWLEDGMENT

The authors would like to thank Prof. D. Landgrebe from Purdue University for providing the AVIRIS image of Indian Pines and Prof. Gamba from the University of Pavia for providing the ROSIS data set. They would also like to thank the Editors and the anonymous reviewers for their detailed comments and suggestions, which greatly helped improve the clarity and presentation of this manuscript.

## REFERENCES

- [1] J. M. Bioucas-Dias, A. Plaza, G. Camps-Valls, P. Scheunders, N. M. Nasrabadi, and J. Chanussot, "Hyperspectral remote sensing data analysis and future challenges," *IEEE Geosci. Remote Sens. Mag.*, vol. 1, no. 2, pp. 6–36, Jun. 2013.
- [2] J. Xia, J. Chanussot, P. Du, and X. He, "Rotation-based support vector machine ensemble in classification of hyperspectral data with limited training samples," *IEEE Trans. Geosci. Remote Sens.*, vol. 54, no. 3, pp. 1519–1531, Mar. 2016.
- [3] W. Liao, M. Dalla Mura, J. Chanussot, and A. Pizurica, "Fusion of spectral and spatial information for classification of hyperspectral remote-sensed imagery by local graph," *IEEE J. Sel. Top. Appl. Earth Observ. Remote Sens.*, vol. 9, no. 2, pp. 583–594, Feb. 2016.
- [4] J. M. Bioucas-Dias *et al.*, "Hyperspectral unmixing overview: Geometrical, statistical, and sparse regression-based approaches," *IEEE J. Sel. Top. Appl. Earth Observ. Remote Sens.*, vol. 5, no. 2, pp. 354–379, Apr. 2012.
- [5] S. Yang, Z. Shi, and W. Tang, "Robust hyperspectral image target detection using an inequality constraint," *IEEE Trans. Geosci. Remote Sens.*, vol. 53, no. 6, pp. 3389–3404, Jun. 2015.
- [6] S. B. Serpico and L. Bruzzone, "A new search algorithm for feature selection in hyperspectral remote sensing images," *IEEE Trans. Geosci. Remote Sens.*, vol. 39, no. 7, pp. 1360–1367, Jul. 2001.
- [7] A. Martínez-Usó, F. Pla, J. M. Sotoca, and P. García-Sevilla, "Clustering-based hyperspectral band selection using information measures," *IEEE Trans. Geosci. Remote Sens.*, vol. 45, no. 12, pp. 4158–4171, Dec. 2007.
- [8] Y. Qian, F. Yao, and S. Jia, "Band selection for hyperspectral imagery using affinity propagation," *IET Comput. Vis.*, vol. 3, no. 4, pp. 213–222, 2009.
- [9] S. Jia, Z. Ji, Y. Qian, and L. Shen, "Unsupervised band selection for hyperspectral imagery classification without manual band removal," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 5, no. 2, pp. 531–543, Apr. 2012.
- [10] S. Jia, G. Tang, J. Zhu, and Q. Li, "A novel ranking-based clustering approach for hyperspectral band selection," *IEEE Trans. Geosci. Remote Sens.*, vol. 54, no. 1, pp. 88–102, Jan. 2016.
- [11] Y. Yuan, J. Lin, and Q. Wang, "Dual-clustering-based hyperspectral band selection by contextual analysis," *IEEE Trans. Geosci. Remote Sens.*, vol. 54, no. 3, pp. 1431–1445, Mar. 2016.
- [12] J. Feng, L. Jiao, F. Liu, T. Sun, and X. Zhang, "Mutual-information-based semi-supervised hyperspectral band selection with high discrimination, high information, and low redundancy," *IEEE Trans. Geosci. Remote Sens.*, vol. 53, no. 5, pp. 2956–2969, May 2015.
- [13] M. A. Hossain, X. Jia, and J. A. Benediktsson, "One-class oriented feature selection and classification of heterogeneous remote sensing images," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 9, no. 4, pp. 1606–1612, Apr. 2016.
- [14] Y. Yuan, G. Zhu, and Q. Wang, "Hyperspectral band selection by multi-task sparsity pursuit," *IEEE Trans. Geosci. Remote Sens.*, vol. 53, no. 2, pp. 631–644, Feb. 2015.
- [15] P. Ghamisi, M. S. Couceiro, and J. A. Benediktsson, "A novel feature selection approach based on FODPSO and SVM," *IEEE Trans. Geosci. Remote Sens.*, vol. 53, no. 5, pp. 2935–2947, May 2015.
- [16] H. Su, B. Yong, and Q. Du, "Hyperspectral band selection using improved firefly algorithm," *IEEE Geosci. Remote Sens. Lett.*, vol. 13, no. 1, pp. 68–72, Jan. 2016.
- [17] I. T. Jolliffe, *Principal Component Analysis*. New York, NY, USA: Springer-Verlag, 2002.
- [18] A. Dadon, E. Ben-Dor, and A. Karnieli, "Use of derivative calculations and minimum noise fraction transform for detecting and correcting the spectral curvature effect (smile) in Hyperion images," *IEEE Trans. Geosci. Remote Sens.*, vol. 48, no. 6, pp. 2603–2612, Mar. 2010.
- [19] L. M. Bruce, C. H. Koger, and J. Li, "Dimensionality reduction of hyperspectral data using discrete wavelet transform feature extraction," *IEEE Trans. Geosci. Remote Sens.*, vol. 40, no. 10, pp. 2331–2338, Oct. 2002.
- [20] S. Jia, L. Shen, and Q. Li, "Gabor feature-based collaborative representation for hyperspectral imagery classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 53, no. 2, pp. 1118–1129, Feb. 2015.
- [21] Z. He, Q. Wang, Y. Shen, and M. Sun, "Kernel sparse multitask learning for hyperspectral image classification with empirical mode decomposition and morphological wavelet-based features," *IEEE Trans. Geosci. Remote Sens.*, vol. 52, no. 8, pp. 5150–5163, Aug. 2014.
- [22] J. Zabalza, J. Ren, Z. Wang, S. Marshall, and J. Wang, "Singular spectrum analysis for effective feature extraction in hyperspectral imaging," *IEEE Geosci. Remote Sens. Lett.*, vol. 11, no. 11, pp. 1886–1890, Nov. 2014.
- [23] B. Hou, X. Zhang, Q. Ye, and Y. Zheng, "A novel method for hyperspectral image classification based on Laplacian eigenmap pixels distribution-flow," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 6, no. 3, pp. 1602–1618, Jun. 2013.
- [24] L. Zhang, L. Zhang, D. Tao, X. Huang, and B. Du, "Hyperspectral remote sensing image subpixel target detection based on supervised metric learning," *IEEE Trans. Geosci. Remote Sens.*, vol. 52, no. 8, pp. 4955–4965, Aug. 2014.

- [25] L. Ma, X. Zhang, X. Yu, and D. Luo, "Spatial regularized local manifold learning for classification of hyperspectral images," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 9, no. 2, pp. 609–624, Feb. 2016.
- [26] P. Quesada-Barriuso, F. Arguello, D. B. Heras, and J. A. Benediktsson, "Wavelet-based classification of hyperspectral images using extended morphological profiles on graphics processing units," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 8, no. 6, pp. 2962–2970, Jun. 2015.
- [27] M. Fauvel, Y. Tarabalka, J. A. Benediktsson, J. Chanussot, and J. C. Tilton, "Advances in spectral–spatial classification of hyperspectral images," *Proc. IEEE*, vol. 101, no. 3, pp. 652–675, Mar. 2013.
- [28] B. Song *et al.*, "Remotely sensed image classification using sparse representations of morphological attribute profiles," *IEEE Trans. Geosci. Remote Sens.*, vol. 52, no. 8, pp. 5122–5136, Aug. 2014.
- [29] B. Demir and S. Erturk, "Empirical mode decomposition of hyperspectral images for support vector machine classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 48, no. 11, pp. 4071–4084, Nov. 2010.
- [30] E. T. Gormus, N. Canagarajah, and A. Achim, "Dimensionality reduction of hyperspectral images using empirical mode decompositions and wavelets," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 5, no. 6, pp. 1821–1830, Dec. 2012.
- [31] A. Erturk, M. K. Gullu, and S. Erturk, "Hyperspectral image classification using empirical mode decomposition with spectral gradient enhancement," *IEEE Trans. Geosci. Remote Sens.*, vol. 51, no. 5, pp. 2787–2798, May 2013.
- [32] J. Zabalza *et al.*, "Novel two-dimensional singular spectrum analysis for effective feature extraction and data classification in hyperspectral imaging," *IEEE Trans. Geosci. Remote Sens.*, vol. 53, no. 8, pp. 4418–4433, Aug. 2015.
- [33] Y. Qian, M. Ye, and J. Zhou, "Hyperspectral image classification based on structured sparse logistic regression and three-dimensional wavelet texture features," *IEEE Trans. Geosci. Remote Sens.*, vol. 51, no. 4, pp. 2276–2291, Apr. 2013.
- [34] T. Lin and S. Bourennane, "Hyperspectral image processing by jointly filtering wavelet component tensor," *IEEE Trans. Geosci. Remote Sens.*, vol. 51, no. 6, pp. 3529–3541, Jun. 2013.
- [35] S. Jia, Z. Zhu, L. Shen, and Q. Li, "A two-stage feature selection framework for hyperspectral image classification using few labeled samples," *IEEE J. Sel. Top. Appl. Earth Observ. Remote Sens.*, vol. 7, no. 4, pp. 1023–1035, Apr. 2014.
- [36] L. Zhang, L. Zhang, D. Tao, and X. Huang, "Tensor discriminative locality alignment for hyperspectral image spectral–spatial feature extraction," *IEEE Trans. Geosci. Remote Sens.*, vol. 51, no. 1, pp. 242–256, Jan. 2013.
- [37] F. Tsai and J. S. Lai, "Feature extraction of hyperspectral image cubes using three-dimensional gray-level cooccurrence," *IEEE Trans. Geosci. Remote Sens.*, vol. 51, no. 6, pp. 3504–3513, Jun. 2013.
- [38] Z. Zhong *et al.*, "Discriminant tensor spectral–spatial feature extraction for hyperspectral image classification," *IEEE Geosci. Remote Sens. Lett.*, vol. 12, no. 5, pp. 1028–1032, May 2015.
- [39] V. N. Vapnik, *The Nature of Statistical Learning Theory*. New York, NY, USA: Springer-Verlag, 1995.
- [40] F. Melgani and L. Bruzzone, "Classification of hyperspectral remote sensing images with support vector machines," *IEEE Trans. Geosci. Remote Sens.*, vol. 42, no. 8, pp. 1778–1790, Aug. 2004.
- [41] P. Ramzi, F. Samadzadegan, and P. Reinartz, "Classification of hyperspectral data using an AdaBoostSVM technique applied on band clusters," *IEEE J. Sel. Top. Appl. Earth Observ. Remote Sens.*, vol. 7, no. 6, pp. 2066–2079, Jun. 2014.
- [42] Y. Chen, N. M. Nasrabadi, and T. D. Tran, "Hyperspectral image classification via kernel sparse representation," *IEEE Trans. Geosci. Remote Sens.*, vol. 51, no. 1, pp. 217–231, Jan. 2013.
- [43] X. Sun, N. M. Nasrabadi, and T. D. Tran, "Task-driven dictionary learning for hyperspectral image classification with structured sparsity constraints," *IEEE Trans. Geosci. Remote Sens.*, vol. 53, no. 8, pp. 4457–4471, Aug. 2015.
- [44] Z. Wu, Q. Wang, A. Plaza, J. Li, J. Liu, and Z. Wei, "Parallel implementation of sparse representation classifiers for hyperspectral imagery on GPUs," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 8, no. 6, pp. 2912–2925, Jun. 2015.
- [45] M. Cui and S. Prasad, "Class-dependent sparse representation classifier for robust hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 53, no. 5, pp. 2683–2695, May 2015.
- [46] E. Zhang, L. Jiao, X. Zhang, H. Liu, and S. Wang, "Class-level joint sparse representation for multifeature-based hyperspectral image classification," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, doi: 10.1109/JSTARS.2016.2522182, to be published.
- [47] M. M. Crawford, D. Tuia, and H. L. Yang, "Active learning: Any value for classification of remotely sensed data?" *Proc. IEEE*, vol. 101, no. 3, pp. 593–608, Mar. 2013.
- [48] Y. Zhang, H. L. Yang, S. Prasad, E. Pasolli, J. Jung, and M. Crawford, "Ensemble multiple kernel active learning for classification of multisource remote sensing data," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 8, no. 2, pp. 845–858, Feb. 2015.
- [49] G. Camps-Valls, L. Gomez-Chova, J. Muñoz-Marí, J. Vila-Francés, and J. Calpe-Maravilla, "Composite kernels for hyperspectral image classification," *IEEE Geosci. Remote Sens. Lett.*, vol. 3, no. 1, pp. 93–97, Jan. 2006.
- [50] D. Tuia, G. Camps-Valls, G. Matasci, and M. Kanevski, "Learning relevant image features with multiple-kernel classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 48, no. 10, pp. 3780–3791, Oct. 2010.
- [51] Y. Gu, Q. Wang, H. Wang, D. You, and Y. Zhang, "Multiple kernel learning via low-rank nonnegative matrix factorization for classification of hyperspectral imagery," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 8, no. 6, pp. 2739–2751, Jun. 2015.
- [52] Z. He and J. Li, "Multiple data-dependent kernel for classification of hyperspectral images," *Expert Syst. Appl.*, vol. 42, no. 3, pp. 1118–1135, 2015.
- [53] Y. Gu, T. Liu, X. Jia, J. A. Benediktsson, and J. Chanussot, "Nonlinear multiple kernel learning with multiple-structure-element extended morphological profiles for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 54, no. 6, pp. 3235–3247, Jun. 2016.
- [54] Y. Tarabalka, M. Fauvel, J. Chanussot, and J. A. Benediktsson, "SVM- and MRF-based method for accurate classification of hyperspectral images," *IEEE Geosci. Remote Sens. Lett.*, vol. 7, no. 4, pp. 736–740, Oct. 2010.
- [55] U. Srinivas, Y. Chen, V. Monga, N. M. Nasrabadi, and T. D. Tran, "Exploiting sparsity in hyperspectral image classification via graphical models," *IEEE Geosci. Remote Sens. Lett.*, vol. 10, no. 3, pp. 505–509, May 2013.
- [56] C. Chen, N. Chen, and J. Peng, "Nearest regularized joint sparse representation for hyperspectral image classification," *IEEE Geosci. Remote Sens. Lett.*, vol. 13, no. 3, pp. 424–428, Mar. 2016.
- [57] Y. Chen, N. M. Nasrabadi, and T. D. Tran, "Hyperspectral image classification using dictionary-based sparse representation," *IEEE Trans. Geosci. Remote Sens.*, vol. 49, no. 10, pp. 3973–3985, Oct. 2011.
- [58] X. Sun, Q. Qu, N. M. Nasrabadi, and T. D. Tran, "Structured priors for sparse-representation-based hyperspectral image classification," *IEEE Geosci. Remote Sens. Lett.*, vol. 11, no. 7, pp. 1235–1239, Jul. 2014.
- [59] F. de Morsier, M. Borgeaud, V. Gass, J. P. Thiran, and D. Tuia, "Kernel low-rank and sparse graph for unsupervised and semi-supervised classification of hyperspectral images," *IEEE Trans. Geosci. Remote Sens.*, vol. 54, no. 6, pp. 3410–3420, Jun. 2016.
- [60] A. Soltani-Farani, H. R. Rabiee, and S. A. Hosseini, "Spatial-aware dictionary learning for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 53, no. 1, pp. 527–541, Jan. 2015.
- [61] P. Du, Z. Xue, J. Li, and A. Plaza, "Learning discriminative sparse representations for hyperspectral image classification," *IEEE J. Sel. Top. Appl. Earth Observ. Remote Sens.*, vol. 9, no. 6, pp. 1089–1104, Sep. 2015.
- [62] J. Li, H. Zhang, L. Zhang, X. Huang, and L. Zhang, "Joint collaborative representation with multitask learning for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 52, no. 9, pp. 5923–5936, Sep. 2014.
- [63] J. Li, H. Zhang, and L. Zhang, "Superpixel-level multitask joint sparse representation for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 53, no. 10, pp. 5338–5351, Oct. 2015.
- [64] Z. Wu, N. E. Huang, and X. Chen, "The multi-dimensional ensemble empirical mode decomposition method," *Adv. Adap. Data Anal.*, vol. 1, no. 3, pp. 339–372, 2009.
- [65] J. Riffi, A. M. Mahraz, A. Abbad, and H. Tairi, "3D extension of the fast and adaptive bidimensional empirical mode decomposition," *Multidimensional Syst. Signal Process.*, vol. 26, pp. 823–834, 2015.
- [66] N. E. Huang *et al.*, "The empirical mode decomposition and the Hilbert spectrum for nonlinear and non-stationary time series analysis," *Proc. Roy. Soc. London A*, vol. 454, no. 1971, pp. 903–995, Mar. 1998.
- [67] S. Siskos, S. Vlassis, and I. Pitas, "Analog implementation of fast min/max filtering," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 45, no. 7, pp. 913–918, Jul. 1998.
- [68] X.-T. Yuan, X. Liu, and S. Yan, "Visual classification with multi-task joint sparse representation," *IEEE Trans. Image Process.*, vol. 21, no. 10, pp. 4349–4360, Oct. 2012.
- [69] S. M. Bhuiyan, R. R. Adhami, and J. F. Khan, "Fast and adaptive bidimensional empirical mode decomposition using order-statistics filter based envelope estimation," *EURASIP J. Adv. Signal Process.*, vol. 2008, no. 1, pp. 1–18, Jan. 2008.



**Zhi He** received the B.S., M.S., and Ph.D. degrees in control science and engineering from the Harbin Institute of Technology, Harbin, China, in 2009, 2011, and 2015, respectively.

She is currently a Lecturer with the School of Geography and Planning, Sun Yat-sen University, Guangzhou, China. Her research interests include Hilbert–Huang transform, sparse representation, and their application to hyperspectral classification.



**Lin Liu** received the B.S. degree in geography and the M.S. degree in remote sensing and cartography from Peking University, Beijing, China, in 1984 and 1987, respectively, and the Ph.D. degree in geography with a specialization in geographic information science from The Ohio State University, Columbus, OH, USA, in 1994.

He is currently a Professor of geography with the University of Cincinnati, Cincinnati, OH, and Sun Yat-sen University, Guangzhou, China. He has published over 100 articles and led multiple national and international research projects. His research interests include GIS, remote sensing, and their applications.



**Jun Li** (M'13) received the B.S. degree in geographic information systems from Hunan Normal University, Changsha, China, in 2004; the M.E. degree in remote sensing from Peking University, Beijing, China, in 2007; and the Ph.D. degree in electrical engineering from the Instituto de Telecomunicações, Instituto Superior Técnico (IST), Universidade Técnica de Lisboa, Lisbon, Portugal, in 2011.

From 2007 to 2011, she was a Marie Curie Research Fellow with the Departamento de Engenharia Electrotécnica e de Computadores and the Instituto de Telecomunicações, IST, Universidade Técnica de Lisboa, in the framework of the European Doctorate for Signal Processing (SIGNAL). She has also been actively involved in the Hyperspectral Imaging Network, which is a Marie Curie Research Training Network involving 15 partners in 12 countries and is intended to foster research, training, and cooperation on hyperspectral imaging at the European level. Since 2011, she has been a Postdoctoral Researcher with the Hyperspectral Computing Laboratory, Department of Technology of Computers and Communications, Escuela Politécnica, University of Extremadura, Cáceres, Spain. She is currently a Professor with Sun Yat-sen University, Guangzhou, China. Her research interests include hyperspectral image classification and segmentation, spectral unmixing, signal processing, and remote sensing.

Dr. Li is an Associate Editor of the IEEE JOURNAL OF SELECTED TOPICS IN APPLIED EARTH OBSERVATIONS AND REMOTE SENSING. She has been a reviewer of several journals, including the IEEE TRANSACTIONS ON GEOSCIENCE AND REMOTE SENSING, the IEEE GEOSCIENCE AND REMOTE SENSING LETTERS, *Pattern Recognition*, *Optical Engineering*, the *Journal of Applied Remote Sensing*, and *Inverse Problems and Imaging*.



**Kai Liu** received the B.S. degree from Chang'an University, Xi'an, China, in 2002 and the Ph.D. degree from the Guangzhou Institute of Geochemistry, Chinese Academy of Sciences, Guangzhou, China, in 2007.

He is currently an Associate Professor with the School of Geography and Planning, Sun Yat-sen University, Guangzhou. His research interests focus on wetlands, particularly on mangrove and dyke pond systems by using remote sensing and geographic information science technologies.



**Li Zhuo** received the Ph.D. degree in natural geography from Beijing Normal University, Beijing, China, in 2005.

In July 2005, she joined the School of Geography and Planning, Sun Yat-sen University, Guangzhou, China, where she is currently an Associate Professor. Her research interests are in the field of remote sensing of environment resources, particularly urban remote sensing, geographic modeling, disaster risk management, and optimization methods.