IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 20 June 2023, accepted 17 July 2023, date of publication 24 July 2023, date of current version 31 July 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3297630

RESEARCH ARTICLE

Tensor Robust CUR for Compression and Denoising of Hyperspectral Data

MOHAMMAD M. SALUT, (Graduate Student Member, IEEE), AND DAVID V. ANDERSON, (Senior Member, IEEE)
School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA 30332, USA
Corresponding author: Mohammad M. Salut (mms30@gatech.edu)

ABSTRACT Hyperspectral images are often contaminated with noise which degrades the quality of data. Recently, tensor robust principal component analysis (TRPCA) has been utilized to remove noise from hyperspectral images, improving classification accuracy. However, the high dimensionality and size of hyperspectral data present computational challenges both in terms of storage and processing power, especially in the case of TRPCA. The situation is exacerbated when the data is too large to fit in available memory. In this paper, we propose a tensor-robust CUR (TRCUR) algorithm for hyperspectral data compression and denoising. We heavily downsample the input hyperspectral image to form small subtensors; and perform TRPCA on the small subtensors. The desired hyperspectral image is recovered by combining the low-rank solution of the subtensors using tensor CUR reconstruction. We provide a theoretical guarantee to show that the desired low-rank tensor can be exactly recovered using our proposed TRCUR method. Numerical experiments indicate that our method is up to 14 times faster than performing TRPCA on the original input data while maintaining the classification accuracy.

INDEX TERMS Low-rank tensor recovery, tensor robust CUR, tensor robust PCA, hyperspectral image denoising, noisy hyperspectral image classification.

I. INTRODUCTION

Hyperspectral images have been widely used in many remote sensing applications e.g. classification [1], [2], [3], [4], [5], [6], [7], [8], [9], [10], [11], [12], unmixing [13], [14], [15], [16], [17], and target detection [18], [19], [20]. Often, hyperspectral images are contaminated with noise [21], [22]. Therefore, recovering the desired information is of great interest. Typically, a hyperspectral image consists of up to several hundred spectral bands, exhibiting high correlation across spectra. Principal component analysis (PCA) has been widely used for spectral dimensionality reduction [23], [24], [25], [26], [27], [28], [29], [30], [31], [32], [33]. A hyperspectral image can be naturally represented by a 3-way tensor so, to perform PCA, a hyperspectral data cube is first converted into a data matrix by vectorizing spectral bands and stacking them as columns of a matrix. Then, singular value decomposition (SVD) is performed on the data matrix to obtain the dominant singular values and vectors. Farrell et al. [23] used PCA to investigate the effect of dimension reduction on detection of difficult targets. Chen et al. [24] proposed a denoising method based on PCA and wavelet shrinkage for high signal-to-noise ratio (SNR) hyperspectral images. While PCA has proven effective, it has some disadvantages when applied to hyperspectral imagery. For example, some spectral bands can have high noise levels, and PCA is sensitive to gross noise and outliers. Moreover, vectorization of hyperspectral data breaks the intrinsic structure of data leading to reduced algorithm effectiveness.

To better exploit the spatial and spectral structures in data, several multidimensional noise reduction techniques have been proposed to effectively reduce noise in hyperspectral data [34], [35], [36], [37], [38], [39]. Recently, Sun et al. [37] used tensor robust PCA (TRPCA) to improve hyperspectral image classification accuracy. TRPCA decomposes a noisy hyperspectral image into a low-rank tensor which contains the desired hyperspectral image, and a sparse tensor which contains noise. Results [37] indicate that TRPCA can

The associate editor coordinating the review of this manuscript and approving it for publication was Qingli Li.

effectively remove noise from hyperspectral images. Lu et al. [40] showed that the resulting TRPCA using tensor nuclear norm can exactly recover the low-rank and sparse tensors from their sum.

Hyperspectral satellites transmit massive amount of high-dimensional data. For example, the NASA's Hyperspectral Infrared Imager (HyspIRI) transmits a volume of 4.5 TBytes per day [41]. TRPCA algorithms, despite their great performance, require large memory and their complexity grows with batch size. The computational complexity of performing TRPCA on hypercube $\mathcal{X} \in \mathbb{R}^{m\times m\times n}$ is $O\left(m^2 n \log n + m^3 n\right)$ flops per iteration [40] which is prohibitive for many applications. The situation becomes particularly critical when data does not fit in available memory.

In this paper, we propose a multi-dimensional divide-and-conquer framework to drastically reduce the computational complexity of TRPCA. We heavily downsample the large input data tensor to obtain small subtensors, and then discard the original data. Therefore, our method is memory-efficient. We perform TRPCA on the small subtensors. The low-rank solution of these subtensors is utilized to recover the desired hyperspectral image via tensor CUR reconstruction. The computational complexity of performing our tensor robust CUR method on hypercube $\mathcal{X} \in \mathbb{R}^{m\times m\times n}$ is $O\left(4mrn \log n + (3m + r)r^2 n + m^2 rn\right)$ flops per iteration where $r \ll m$. The main contributions of this work are as follows.

1) We develop a fast and memory-efficient tensor robust CUR (TRCUR) algorithm for large-scale data sets.
2) We provide the theoretical guarantee to show that the low-rank component can be exactly recovered using our proposed TRCUR method.
3) We demonstrate the accuracy and computational benefits of our method on three hyperspectral data sets. Results indicate that our method can significantly speed up the computation, and effectively reduce noise in hyperspectral images.

II. RELATED WORK

Robust PCA (RPCA) algorithms have been used in many remote sensing applications [37], [42], [43], [44], [45]. Rambhatla et al. [43] proposed a dictionary-based RPCA algorithm for target localization in hyperspectral imaging. Liu et al. [44] developed a log-based RPCA algorithm to remove noise in hyperspectral images. Lee et al. [42] developed a method for tree species classification using RPCA for feature extraction. Recently, Cai et al. [46], [47] integrated CUR decomposition with RPCA to speed up the computation and reduce memory requirements for background subtraction in videos. All the aforementioned methods [42], [43], [44], [46], [47] can only operate on two-way (matrix) data, and cannot directly be applied to hypercubes. Vectorization of hyperspectral data ignores the spatial correlations among neighboring pixels. It is widely accepted that utilizing both spatial and spectral structures in hyperspectral data can significantly improve classification accuracy [41].

Sun et al. [37] proposed a *tensor* RPCA method to exploit the spatial and spectral structures in hyperspectral data. TRPCA [37] can effectively reduce noise in hyperspectral data, improving the classification accuracy. However, RPCA and TRPCA algorithms [37], [40], [48] require the data to be stored in memory for the SVD computation. Most (T)RPCA algorithms require many iterations to converge. Therefore, these methods require high memory overhead and are computationally prohibitive for large-scale datasets. Recently, Salut et al. [49] employed randomized techniques to accelerate TRPCA. The authors [49] utilized their method for noisy hyperspectral image classification. Results [49] indicate that their randomized TRPCA method is significantly faster than performing TRPCA via the full T-SVD while maintaining classification accuracy. However, randomized TRPCA [49] requires at least two passes over the input data per iteration. When the input data is too large to fit in fast memory, this aspect can be critical since the cost of memory access may become higher than the theoretical costs of the algorithm.

Mahoney et al. [50] proposed tensor CUR decomposition for compression based on the Tucker decomposition. The Tucker decomposition decomposes a tensor into a small core tensor multiplied by a set of basis matrices [51], [52]. However, their method [50] is not robust to noise and outliers. Cai et al. [53] proposed a tensor-robust CUR algorithm for background subtraction in videos based on Tucker decomposition. Recently, Kilmer et al. [54], [55], [56] proposed a new tensor decomposition technique based on T-product. The main advantage of T-product-based TRPCA e.g. [37], [40], and [49] over robust Tucker-based methods e.g. [53] is the use of the tensor nuclear norm instead of the Tucker-rank which preserves the data structures among modes [57]. Lu et al. [40] showed that T-product-based TRPCA using tensor nuclear norm can exactly recover the low-rank and sparse tensors from their sum.

Hyperspectral images consist of up to several hundred spectral bands, possessing complex structures. Often hyperspectral images are contaminated with noise, predicting an accurate class label to every pixel in an image is a challenging task. In this paper, we propose a tensor-robust CUR algorithm for compression and denoising of hyperspectral data based on the recently proposed T-product [55], [56]. We sample the large input data tensor to obtain small subtensors, and then discard the original data. We perform TRPCA on the small subtensors resulting in a fast and memory-efficient scheme. We utilize the low-rank solution of these subtensors to recover the desired hyperspectral image via tensor CUR reconstruction. We show that the classification accuracy of our method is still maintained even after significant data compression.

III. PRELIMINARIES

In this paper, scalars are denoted by lowercase letters, e.g., c. Vectors are denoted by boldface lowercase letters, e.g., $\boldsymbol{c}$. Matrices are denoted by capital letters, e.g., C. Third-order tensors are denoted by Euler script letters, e.g., $\mathcal{C} \in \mathbb{R}^{n_1\times n_2\times n_3}$. The j-th lateral slice of $\mathcal{C}$ is denoted by a tensor

$\overrightarrow{\mathcal{C}_j} \equiv \mathcal{C}_{:j:} \equiv \mathcal{C}(:, j, :)$. The k-th frontal slice of tensor $\mathcal{C}$ is denoted by a matrix $C^{(k)} \equiv \mathcal{C}_{::k} \equiv \mathcal{C}(:, :, k)$. The block circulant matrix of tensor $\mathcal{C} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ is define as:

$$\texttt{bcirc}(\mathcal{C}) = \begin{pmatrix} C^{(1)} & C^{(n_3)} & \cdots & C^{(2)} \\ C^{(2)} & C^{(1)} & \cdots & C^{(3)} \\ \vdots & \vdots & \ddots & \vdots \\ C^{(n_3)} & C^{(n_3-1)} & \cdots & C^{(1)} \end{pmatrix} \tag{1}$$

The block circulant matrix can be diagonalized by a discrete Fourier transform (DFT). Using bar notation ($\bar{\cdot}$) to represent the Fourier domain, we denote $\bar{C} \in \mathbb{C}^{n_1 n_3 \times n_2 n_3}$ as the block-diagonal matrix of data tensor $\bar{\mathcal{C}} \in \mathbb{C}^{n_1 \times n_2 \times n_3}$.

$$\begin{aligned} \bar{C} &= \left(F_{n_3} \otimes I_{n_1}\right) \cdot \texttt{bcirc}(\mathcal{C}) \cdot \left(F_{n_3}^H \otimes I_{n_2}\right) \\ &= \texttt{bdiag}\left(\bar{\mathcal{C}}\right) = \begin{pmatrix} \bar{C}^{(1)} & & & \\ & \bar{C}^{(2)} & & \\ & & \ddots & \\ & & & \bar{C}^{(n_3)} \end{pmatrix} \end{aligned} \tag{2}$$

where "$\otimes$" denotes the Kronecker product. F_n is the DFT matrix. F_n^H denotes the conjugate transpose of F_n. "$\cdot$" represents matrix product. $\bar{\mathcal{C}}$ denotes the DFT of data tensor $\mathcal{C}$ which is computed by taking the Fast Fourier Transform (FFT) along each tube, using MATLAB notation:

$$\bar{\mathcal{C}} = \texttt{fft}(\mathcal{C}, [\,], 3) \tag{3}$$

We can matricize a tensor by unfolding its frontal slices.

$$\texttt{unfold}(\mathcal{C}) = \begin{bmatrix} C^{(1)} \\ C^{(2)} \\ \vdots \\ C^{(n_3)} \end{bmatrix} \tag{4}$$

The `fold` operator converts back $\texttt{unfold}(\mathcal{C})$ to its original tensor form.

$$\texttt{fold}(\texttt{unfold}(\mathcal{C})) = \mathcal{C} \tag{5}$$

A. T-PRODUCT

Let $\mathcal{C} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ and $\mathcal{D} \in \mathbb{D}^{n_2 \times n_4 \times n_3}$. The T-product $\mathcal{Y} \in \mathbb{R}^{n_1 \times n_4 \times n_3}$ is defined as [54], [55], [56]

$$\mathcal{Y} = \mathcal{C} * \mathcal{D} = \texttt{fold}\left(\texttt{bcirc}\left(\mathcal{C}\right) \cdot \texttt{unfold}\left(\mathcal{D}\right)\right) \tag{6}$$

Computing the T-product in the original domain is computationally inefficient unless the tensors are sparse. In this paper, we perform the T-product in the Fourier domain.

$$\begin{aligned} &\texttt{unfold}(\mathcal{Y}) \\ &= \texttt{bcirc}\left(\mathcal{C}\right) \cdot \texttt{unfold}\left(\mathcal{D}\right) \\ &= \left(F_{n_3}^H \otimes I_{n_1}\right) \cdot \left(\left(F_{n_3} \otimes I_{n_1}\right) \cdot \texttt{bcirc}(\mathcal{C}) \cdot \left(F_{n_3}^H \otimes I_{n_2}\right)\right) \\ &\quad \cdot \left(F_{n_3} \otimes I_{n_2}\right) \cdot \texttt{unfold}\left(\mathcal{D}\right) \\ &= \left(F_{n_3}^H \otimes I_{n_1}\right) \cdot \bar{\mathcal{C}} \cdot \texttt{unfold}\left(\bar{\mathcal{D}}\right) \end{aligned} \tag{7}$$

By multiplying both sides of (7) with $\left(F_{n_3} \otimes I_{n_1}\right)$, we obtain

$$\bar{Y}^{(i)} = \bar{C}^{(i)} \cdot \bar{D}^{(i)} \tag{8}$$

which is equivalent to multiplying each frontal slice of $\bar{\mathcal{C}}$ with its counterpart in $\bar{\mathcal{D}}$. Due to the conjugate symmetry property of the DFT for real-valued data, we only need to compute $\bar{Z}^{(i)} = \bar{C}^{(i)} \cdot \bar{D}^{(i)}$ for about half of the transformed frontal slices [40]. The T-product is summarized in Algorithm 1.

Algorithm 1 T-Product

Input: $\mathcal{C} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$, $\mathcal{D} \in \mathbb{R}^{n_2 \times n_4 \times n_3}$
Output: $\mathcal{Y} \in \mathbb{R}^{n_1 \times n_4 \times n_3}$

```
C̄ = fft(C, [ ], 3)
D̄ = fft(D, [ ], 3)
for i = 1 : ⌈(n3+1)/2⌉ do
    Ȳ^(i) = C̄^(i) · D̄^(i)
end for
for i = ⌈(n3+1)/2⌉ + 1 : n3 do
    Ȳ^(i) = conj(Ȳ^(n3−i+2))
end for
Y = ifft(Ȳ, [ ], 3)
```

B. CONJUGATE TRANSPOSE

The conjugate transpose of a tensor $\mathcal{L} \in \mathbb{C}^{n_1 \times n_2 \times n_3}$ is obtained by conjugate transposing each of the frontal slices and reversing the order of frontal slices 2 through n_3, $\mathcal{L}^H \in \mathbb{C}^{n_2 \times n_1 \times n_3}$.

C. IDENTITY TENSOR

$\mathcal{I} \in \mathbb{R}^{n_1 \times n_1 \times n_3}$ is an identity tensor having its first frontal slice being the $n_1 \times n_1$ identity matrix, and zeros everywhere else. The Fourier transform of identity tensor ($\mathcal{I}$) along the third dimension is denoted as $\bar{\mathcal{I}}$. Every frontal slice of $\bar{\mathcal{I}}$ is an identity matrix $\bar{\mathcal{I}}^{(i)} = I$ [58].

D. TENSOR PSEUDO-INVERSE

The tensor pseudo-inverse of $\mathcal{L} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ is denoted by $\mathcal{L}^\dagger \in \mathbb{R}^{n_2 \times n_1 \times n_3}$ and is obtained by computing the Moore-Penrose pseudo-inverse of the frontal slices in the frequency domain $\bar{L}^{(i)^\dagger}$ [59]. The tensor pseudo-inverse satisfies the following criteria [60]:

1) $\mathcal{L} * \mathcal{L}^\dagger * \mathcal{L} = \mathcal{L}$
2) $\mathcal{L}^\dagger * \mathcal{L} * \mathcal{L}^\dagger = \mathcal{L}^\dagger$
3) $\left(\mathcal{L} * \mathcal{L}^\dagger\right)^T = \mathcal{L} * \mathcal{L}^\dagger$
4) $\left(\mathcal{L}^\dagger * \mathcal{L}\right)^T = \mathcal{L}^\dagger * \mathcal{L}$

E. TENSOR SPECTRAL NORM

The tensor spectral norm of data tensor $\mathcal{L}$ is defined as [40]:

$$\|\mathcal{L}\| = \|\texttt{bcirc}\left(\mathcal{L}\right)\| = \|\bar{L}\| \tag{9}$$

F. TENSOR CONDITION NUMBER

The tensor condition number of $\mathcal{L} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ is defined as:

$$\begin{aligned} \kappa &= \|\texttt{bcirc}\left(\mathcal{L}\right)\| \|(\texttt{bcirc}\left(\mathcal{L}\right))^\dagger\| \\ &= \|\bar{L}\| \|\bar{L}^\dagger\| = \frac{\sigma_{max}(\bar{L})}{\sigma_{min}(\bar{L})} \end{aligned} \tag{10}$$

where $\sigma_{max}(\bar{L})$ and $\sigma_{min}(\bar{L})$ are the the maximum and minimum nonzero singular values of $\bar{L}$, respectively.

G. ORTHOGONAL TENSOR

A tensor $\mathcal{W} \in \mathbb{R}^{n_1 \times n_1 \times n_3}$ is orthogonal if $\mathcal{W} * \mathcal{W}^T = \mathcal{W}^T * \mathcal{W} = \mathcal{I}$.

H. F-DIAGONAL TENSOR

A tensor is f-diagonal if each frontal slice is a diagonal matrix [54], [55], [56].

I. T-SVD

The T-SVD of $\mathcal{L} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ factorizes $\mathcal{L}$ into the T-product of three tensors $\mathcal{L} = \mathcal{W} * \mathcal{S} * \mathcal{V}^T$ where $\mathcal{W} \in \mathbb{R}^{n_1 \times n_1 \times n_3}$ and $\mathcal{V} \in \mathbb{R}^{n_2 \times n_2 \times n_3}$ are orthogonal tensors and $\mathcal{S} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ is a f-diagonal tensor [40], [61], [62]. Algorithm 2 summarizes T-SVD factorization for real-valued data.

Algorithm 2 T-SVD

Input: $\mathcal{L}$
Output: $\mathcal{W}$, $\mathcal{S}$, $\mathcal{V}$

$\bar{\mathcal{L}} = \texttt{fft}(\mathcal{L}, [\,], 3)$;
for $i = 1 : \lceil \frac{n_3+1}{2} \rceil$ **do**
  $[\bar{W}^{(i)} \quad \bar{S}^{(i)} \quad \bar{V}^{(i)}] = \texttt{svd}\left(\bar{L}^{(i)}\right)$
end for
for $i = \lceil \frac{n_3+1}{2} \rceil + 1 : n_3$ **do**
  $\bar{W}^{(i)} = \texttt{conj}\left(\bar{W}^{(n_3-i+2)}\right)$
  $\bar{S}^{(i)} = \bar{S}^{(n_3-i+2)}$
  $\bar{V}^{(i)} = \texttt{conj}\left(\bar{V}^{(n_3-i+2)}\right)$
end for
$\mathcal{W} = \texttt{ifft}(\bar{\mathcal{W}}, [\,], 3)$
$\mathcal{S} = \texttt{ifft}(\bar{\mathcal{S}}, [\,], 3)$
$\mathcal{V} = \texttt{ifft}(\bar{\mathcal{V}}, [\,], 3)$

J. TENSOR TUBAL RANK

Let $\mathcal{L} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ has the T-SVD $\mathcal{L} = \mathcal{W} * \mathcal{S} * \mathcal{V}^T$. The tensor tubal rank of $\mathcal{L}$ is defined as the number of nonzero singular tubes of $\mathcal{S}$ [40]:

$$\begin{aligned} \texttt{rank}_T(\mathcal{L}) &= \#\{i, \mathcal{S}(i,i,:) \neq \mathbf{0}\} \\ &= \#\{i, \mathcal{S}(i,i,1) \neq 0\} \end{aligned} \tag{11}$$

K. T-CUR

Let $\mathcal{L} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ be a low-rank tensor. $\mathcal{C} \in \mathbb{R}^{n_1 \times |J| \times n_3}$ is a subtensor of $\mathcal{L}$ consisting of those lateral slices of $\mathcal{L}$ indexed by $J \subset \{1, \ldots, n_2\}$. Similarly, $\mathcal{R} \in \mathbb{R}^{|I| \times n_2 \times n_3}$ is a subtensor of $\mathcal{L}$ consisting of those horizontal slices of $\mathcal{L}$ indexed by $I \subset \{1, \ldots, n_1\}$. The small linkage tensor $\mathcal{U} \in \mathbb{R}^{|I| \times |J| \times n_3}$ is the intersection of subtensors $\mathcal{C}$ and $\mathcal{R}$. If $\texttt{rank}_T(\mathcal{U}) = \texttt{rank}_T(\mathcal{L})$, then [60]

$$\mathcal{L} = \mathcal{C} * \mathcal{U}^\dagger * \mathcal{R} \tag{12}$$

Tensor CUR decomposition is also referred to as tensor skeleton decomposition. Algorithm 3 summarizes T-CUR decomposition.

Algorithm 3 T-CUR

Input: $\mathcal{L} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$, $|I|$, $|J|$
Output: $\hat{\mathcal{L}}$

$I \subset \{1, \ldots, n_1\}$
$J \subset \{1, \ldots, n_2\}$
$\mathcal{C} = \mathcal{L}_{:J:}$
$\mathcal{R} = \mathcal{L}_{I::}$
$\bar{\mathcal{C}} = \texttt{fft}(\mathcal{C}, [\,], 3)$
$\bar{\mathcal{R}} = \texttt{fft}(\mathcal{R}, [\,], 3)$
$\bar{\mathcal{U}} = \bar{\mathcal{R}}_{:J:}$
for $i = 1 : \lceil \frac{n_3+1}{2} \rceil$ **do**
  $\hat{\bar{L}}^{(i)} = \bar{C}^{(i)} \cdot \bar{U}^{(i)^\dagger} \cdot \bar{R}^{(i)}$
end for
for $i = \lceil \frac{n_3+1}{2} \rceil + 1 : n_3$ **do**
  $\hat{\bar{L}}^{(i)} = \texttt{conj}\left(\hat{\bar{L}}^{(n_3-i+2)}\right)$
end for
$\hat{\mathcal{L}} = \texttt{ifft}(\hat{\bar{\mathcal{L}}}, [\,], 3)$

L. TENSOR ROBUST PCA

tensor robust PCA decomposes a given data tensor as $\mathcal{X} = \mathcal{L} + \mathcal{E} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$, where $\mathcal{L}$ is a low-rank tensor and $\mathcal{E}$ is a sparse tensor. The low-rank tensor $\mathcal{L}$ is recovered by solving the following convex problem [40]

$$\min_{\mathcal{L},\mathcal{E}} \|\mathcal{L}\|_* + \lambda \|\mathcal{E}\|_1, \quad \text{s.t} \quad \mathcal{X} = \mathcal{L} + \mathcal{E} \tag{13}$$

where $\|\cdot\|_*$ and $\|\cdot\|_1$ denote the tensor nuclear norm and ℓ_1 norm, respectively. λ is the regularization parameter. It has been shown that the low-rank tensor can be exactly recovered with high probability if it satisfies tensor incoherence conditions [40].

M. TENSOR INCOHERENCE CONDITIONS

Let $\mathcal{L} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ be a low-rank tensor of tubal-rank r whose compact T-SVD is given by $\mathcal{W} * \mathcal{S} * \mathcal{V}^T$, where $\mathcal{W} \in \mathbb{R}^{n_1 \times r \times n_3}$ and $\mathcal{S} \in \mathbb{R}^{r \times r \times n_3}$ and $\mathcal{V} \in \mathbb{R}^{n_2 \times r \times n_3}$. $\mathcal{L}$ is $\{\mu_1, \mu_2, \mu_3\}$-incoherent if

$$\max_i \|\mathcal{W}^T * \vec{e_i}\| \leq \sqrt{\frac{\mu_1 r}{n_1 n_3}} \tag{14}$$

$$\max_j \|\mathcal{V}^T * \vec{e_j}\| \leq \sqrt{\frac{\mu_2 r}{n_2 n_3}} \tag{15}$$

$$\|\mathcal{W} * \mathcal{V}^T\|_\infty \leq \sqrt{\frac{\mu_3 r}{n_1 n_2 n_3^2}} \tag{16}$$

for some constants μ_1, μ_2, and μ_3. $\vec{e_i} \in \mathbb{R}^{n_1 \times 1 \times n_3}$ is the tensor standard basis with all entries equal to 0 but its $(i, 1, 1)$-th entry equal to 1. If we let μ_1, μ_2, and μ_3 be the smallest numbers, satisfying eq. (14 − 16), then the joint incoherence parameter μ_3 is the dominant factor in these bounds $\mu_3 \geq \max(\mu_1, \mu_2)$. In some important settings, μ_3 is as large as $\mu_1 \mu_2$ [63].

IV. PROPOSED ALGORITHM

We present our multi-dimensional divide-and-conquer framework to drastically reduce the computational complexity of TRPCA without sacrificing the accuracy. Our proposed algorithm takes as inputs a noisy hyperspectral image $\mathcal{X} = \mathcal{L} + \mathcal{E} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$, and the sample size parameters $|I|$ and $|J|$. A hyperspectral image consists of $n_1 \times n_3$ pixels, and n_2 spectral bands. We are interested in decomposing $\mathcal{X}$ into the desired low-rank hyperspectral image $\mathcal{L}$ and the sparse noise tensor $\mathcal{E}$. In this paper, we consider uniform random sampling of horizontal and lateral slices of data cube $\mathcal{X}$ to produce a skeleton decomposition. Let I be a set of random indices of size $|I|$ uniformly chosen from $\{1, \cdots, n_1\}$. We construct subtensor $\tilde{\mathcal{R}} \in \mathbb{R}^{|I| \times n_2 \times n_3}$ by selecting those horizontal slices of $\mathcal{X}$ indexed by I. Similarly, subtensor $\tilde{\mathcal{C}} \in \mathbb{R}^{n_1 \times |J| \times n_3}$ is obtained by restricting $\mathcal{X}$ to the lateral slices indexed by J. Then, we discard the noisy hyperspectral image $\mathcal{X}$. This reduces memory space from $O(n_1 n_2 n_3)$ to $O\left(|I| n_2 n_3 + n_1 |J| n_3 + |I||J| n_3\right)$ where $|I| \ll n_1$ and $|J| \ll n_2$. Thus, the compression ratio of our proposed method is given by:

$$\text{Compression Ratio} = \frac{n_1 n_2 n_3}{n_1 |J| n_3 + |I||J| n_3 + |I| n_2 n_3} \tag{17}$$

In our numerical experiments; as will be shown later, we selected only $10\% - 20\%$ of the slices of the input data tensor to form the subtensors, resulted in comparable or even improved accuracy when compared to performing TRPCA on the original hyperspectral image. The compression attained using our proposed method indicates that high redundancy intrinsically exists in hyperspectral data.

We apply TRPCA [40] on the noisy subtensors $\tilde{\mathcal{C}} = \mathcal{C} + \mathcal{E}_c$ and $\tilde{\mathcal{R}} = \mathcal{R} + \mathcal{E}_r$ to recover their low-rank components $\mathcal{C}$ and $\mathcal{R}$. Subsequently, we compute the small linkage tensor $\mathcal{U}$ as $\mathcal{U} = \mathcal{R}_{:J:} \in \mathbb{R}^{|I| \times |J| \times n_3}$. To successfully recover $\mathcal{C}$ and $\mathcal{R}$ form their noisy measurements, the subtensors must be incoherent. The desired hyperspectral image is obtained using T-CUR reconstruction $\hat{\mathcal{L}} = \mathcal{C} * \mathcal{U}^{\dagger} * \mathcal{R}$. The computational cost of performing TRPCA [40] on subtensor $\tilde{\mathcal{C}}$ and $\tilde{\mathcal{R}}$ is $O(n_1 |J| n_3 log(n_3) + n_1 |J|^2 n_3)$ and $O(|I| n_2 n_3 log(n_3) + n_2 |I|^2 n_3)$ flops per iteration. The cost of performing the T-CUR reconstruction is $O(n_1 |J| n_3 log(n_3) + |I||J| n_3 log(n_3) + |I| n_2 n_3 log(n_3) + max(|I|, |J|)^2 min(|I|, |J|) n_3 + n_1 |I||J| n_3 + n_1 |I| n_2 n_3)$. Algorithm 4 summarizes our tensor robust CUR algorithm.

Algorithm 4 Tensor Robust CUR Algorithm

Input: $\mathcal{X} = \mathcal{L} + \mathcal{E} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$, $|I|$, $|J|$
Output: $\mathcal{L}$

1: $I \subset \{1, \ldots, n_1\}$
2: $J \subset \{1, \ldots, n_2\}$
3: $\tilde{\mathcal{C}} = \mathcal{X}_{:J:}$
4: $\tilde{\mathcal{R}} = \mathcal{X}_{I::}$
5: $\mathcal{C} = TRPCA(\tilde{\mathcal{C}})$
6: $\mathcal{R} = TRPCA(\tilde{\mathcal{R}})$
7: $\mathcal{U} = \mathcal{R}_{:J:}$
8: $\hat{\mathcal{L}} = \mathcal{C} * \mathcal{U}^{\dagger} * \mathcal{R}$

A. EXACT RECOVERY GUARANTEE OF TRCUR

We provide the sufficient conditions that guarantee the success of our method. We begin with finding bounds on incoherence of randomly sampled subtensors $\mathcal{C}$, $\mathcal{U}$, and $\mathcal{R}$ in terms of the incoherence and condition number of the original low-rank tensor $\mathcal{L}$. This is similar to the matrix robust CUR approach [47].

Theorem 1: Let $\mathcal{L} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ have tubal-rank r with compact tensor singular value decomposition $\mathcal{L} = \mathcal{W} * \mathcal{S} * \mathcal{V}^T$, satisfying the tensor incoherence conditions (14−16). Let $\mathcal{C} = \mathcal{W}_{\mathcal{C}} * \mathcal{S}_{\mathcal{C}} * \mathcal{V}_{\mathcal{C}}^T$ be a subtensor of $\mathcal{L}$ formed by uniformly and randomly selecting $|J|$ lateral slices of $\mathcal{L}$ such that $\texttt{rank}_T(\mathcal{C}) = \texttt{rank}_T(\mathcal{L})$, and $\beta = \|(\mathcal{V}_{J::})^{\dagger}\|$. Then,

$$\max_i \|\mathcal{W}_{\mathcal{C}}^T * \vec{e_i}\| \leq \sqrt{\frac{\mu_1 r}{n_1 n_3}} \tag{18}$$

$$\max_j \|\mathcal{V}_{\mathcal{C}}^T * \vec{e_j}\| \leq \beta\kappa \sqrt{\frac{\mu_2 r}{n_2 n_3}} \tag{19}$$

$$\|\mathcal{W}_{\mathcal{C}} * \mathcal{V}_{\mathcal{C}}^T\|_{\infty} \leq \beta\kappa \sqrt{\frac{\mu_1 \mu_2 r}{n_1 n_2 n_3^2}} \tag{20}$$

Lemma 1: Let $\mathcal{L} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ satisfy the tensor incoherence conditions (14−16). Let $\mathcal{C} = \mathcal{L}_{:J:}$ be any random subset of the lateral slices of $\mathcal{L}$ such that $\texttt{rank}_T(\mathcal{C}) = \texttt{rank}_T(\mathcal{L})$. Then

$$\max_i \|\mathcal{W}_{\mathcal{C}}^T * \vec{e_i}\| \leq \sqrt{\frac{\mu_1 r}{n_1 n_3}}$$

Proof: Note that $\mathcal{C} = \mathcal{L}_{:J:} = \mathcal{W} * \mathcal{S} * (\mathcal{V}_{J::})^T$. The T-SVD of $\mathcal{S} * (\mathcal{V}_{J::})^T$ is $\hat{\mathcal{W}}_{\mathcal{C}} * \mathcal{S}_{\mathcal{C}} * \mathcal{V}_{\mathcal{C}}^T$. Thus, $\mathcal{C} = \mathcal{W} * \hat{\mathcal{W}}_{\mathcal{C}} * \mathcal{S}_{\mathcal{C}} * \mathcal{V}_{\mathcal{C}}{}^T$, and $\mathcal{W}_{\mathcal{C}} = \mathcal{W} * \hat{\mathcal{W}}_{\mathcal{C}}$. Now,

$$\begin{aligned}
\max_i \|\mathcal{W}_{\mathcal{C}}^T * \vec{e_i}\| &= \max_i \|\hat{\mathcal{W}}_{\mathcal{C}}^T * \mathcal{W}^T * \vec{e_i}\| \\
&= \max_i \|\hat{\bar{W}}_{\mathcal{C}}^H \cdot \bar{W}^H \cdot \vec{e_i}\| = \max_i \|\bar{W}^H \cdot \vec{e_i}\| \\
&= \max_i \|\mathcal{W}^T * \vec{e_i}\|
\end{aligned}$$

Thus,

$$\max_i \|\mathcal{W}_{\mathcal{C}}^T * \vec{e_i}\| \leq \sqrt{\frac{\mu_1 r}{n_1 n_3}}$$

□

Lemma 2: Let $\mathcal{L} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ satisfies the tensor incoherence conditions (14−16). Let $\mathcal{C} = \mathcal{L}_{:J:}$ be any random subset of the lateral slices of $\mathcal{L}$ such that $\texttt{rank}_T(\mathcal{C}) = \texttt{rank}_T(\mathcal{L})$. Then

$$\max_j \|\mathcal{V}_{\mathcal{C}}^T * \vec{e_j}\| \leq \beta\kappa \sqrt{\frac{\mu_2 r}{n_1 n_3}}$$

Proof: We have $\mathcal{C} = \mathcal{W}_\mathcal{C} * \mathcal{S}_\mathcal{C} * \mathcal{V}_\mathcal{C}^T$, and thus $\mathcal{V}_\mathcal{C}^T = \mathcal{S}_\mathcal{C}^{-1} * \mathcal{W}_\mathcal{C}^T * \mathcal{C}$.

$$\begin{aligned}
\max_j \|\mathcal{V}_\mathcal{C}^T * \overrightarrow{e_j}\| &= \max_j \|\mathcal{S}_\mathcal{C}^{-1} * \mathcal{W}_\mathcal{C}^T * \mathcal{C} * \overrightarrow{e_j}\| \\
&= \max_j \|\bar{S}_\mathcal{C}^{-1} \cdot \bar{W}_\mathcal{C}^H \cdot \bar{C} \cdot \overrightarrow{e_j}\| \\
&\le \|\bar{S}_\mathcal{C}^{-1}\| \|\bar{W}_\mathcal{C}^H\| \|\bar{C} \cdot \overrightarrow{e_j}\| \\
&= \|\bar{C}^\dagger\| \|\bar{W} \cdot \bar{S} \cdot \bar{V}_{J::}^H \cdot \overrightarrow{e_j}\| \\
&= \|\bar{V}_{J::}^\dagger \cdot \bar{S}^{-1} \cdot \bar{W}^\dagger\| \|\bar{W} \cdot \bar{S} \cdot \bar{V}_{J::}^H \cdot \overrightarrow{e_j}\| \\
&\le \|\bar{V}_{J::}^\dagger\| \|\bar{S}^{-1}\| \|\bar{W}^\dagger\| \|\bar{W}\| \|\bar{S}\| \|\bar{V}_{J::}^H \cdot \overrightarrow{e_j}\| \\
&= \beta \|\bar{L}^\dagger\| \|\bar{L}\| \|\bar{V}_{J::}^H \cdot \overrightarrow{e_j}\| \\
&= \beta\kappa \|\mathcal{V}_{J::}^T * \overrightarrow{e_j}\| \le \beta\kappa \|\mathcal{V}^T * \overrightarrow{e_j}\|
\end{aligned}$$

Thus,

$$\max_j \|\mathcal{V}_\mathcal{C}^T * \overrightarrow{e_j}\| \le \beta\kappa \sqrt{\frac{\mu_2 r}{n_1 n_3}}$$

□

Lemma 3: Let $\mathcal{L} = \mathcal{W} * \mathcal{S} * \mathcal{V}^T$ be a $n_1 \times n_2 \times n_3$ with tubal-rank r, satisfying the tensor incoherence conditions (14−16). Let $\mathcal{C} = \mathcal{L}_{:J:}$ be any random subset of the lateral slices of $\mathcal{L}$ such that $\mathtt{rank}_T(\mathcal{C}) = \mathtt{rank}_T(\mathcal{L})$. Then,

$$\|\mathcal{W}_\mathcal{C} * \mathcal{V}_\mathcal{C}^T\|_\infty \le \beta\kappa \sqrt{\frac{\mu_1 \mu_2 r}{n_1 n_2 n_3^2}}$$

Proof: According to Lemma 1 and 2, we have $\mu_{\mathcal{C}_1} \le \mu_1$ and $\mu_{\mathcal{C}_2} \le \beta^2\kappa^2\mu_2$. The proof follows

$$\|\mathcal{W}_\mathcal{C} * \mathcal{V}_\mathcal{C}^T\|_\infty = \|\bar{W}_\mathcal{C} \cdot \bar{V}_\mathcal{C}^H\|_\infty \le \beta\kappa \sqrt{\frac{\mu_1 \mu_2 r}{n_1 n_2 n_3^2}}$$

□

Theorem 2: Let $\mathcal{L} = \mathcal{W} * \mathcal{S} * \mathcal{V}^T$ be a $n_1 \times n_2 \times n_3$ tensor with tubal-rank r, satisfying the tensor incoherence conditions (14−16). For a positive parameter γ_1, select $|J| = \gamma_1 \mu_2 r$ lateral slices of $\mathcal{L}$ uniformly and randomly without replacement to form subtensor $\mathcal{C} = \mathcal{L}_{:J:} = \mathcal{W} * \mathcal{S} * \mathcal{V}_{J::}^T$. Then, $\beta = \|\mathcal{V}_{J::}^\dagger\|$ satisfies eq (19) and the following condition:

$$\beta \le \sqrt{\frac{n_2}{(1-\delta)|J|}} \tag{21}$$

with probability of at least $1 - rn_3 \cdot \left(\frac{e^{-\delta}}{(1-\delta)^{(1-\delta)}}\right)^{\gamma_1 r n_3}$ for all $\delta \in [0, 1)$.

Proof: The block diagonal matrix $\mathtt{bdiag}(\bar{\mathcal{V}}) = \bar{V} \in \mathbb{C}^{n_2 n_3 \times r n_3}$ has orthonormal columns. We draw a random subset J from $\{1, \cdots, n_2\}$ without replacement to construct subtensor $\mathcal{V}_{J::} \in \mathbb{R}^{|J| \times r \times n_3}$. Then, $\bar{V}_{J::} \in \mathbb{C}^{|J| n_3 \times r n_3}$,

$$\|\mathcal{V}_{J::}\| = \|\bar{V}_{J::}\| = \left\| \begin{pmatrix} \bar{V}_{J:1} & & & \\ & \bar{V}_{J:2} & & \\ & & \ddots & \\ & & & \bar{V}_{J:n_3} \end{pmatrix} \right\|$$

The proof follows Lemma 3.4 of [64].

$$\sqrt{\frac{(1-\delta)|J|n_3}{n_2 n_3}} \le \sigma_{min}\left(\bar{V}_{J::}\right)$$

Note that $\left\|\mathcal{V}_{J::}^\dagger\right\| = \left\|\bar{V}_{J::}^\dagger\right\| = \left(\sigma_{min}(\bar{V}_{J::})\right)^{-1}$.

$$\beta = \left\|\mathcal{V}_{J::}^\dagger\right\| \le \sqrt{\frac{n_2}{(1-\delta)|J|}}$$

□

Theorem 3: Let $\mathcal{L} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ have tubal-rank r with compact tensor singular value decomposition $\mathcal{L} = \mathcal{W} * \mathcal{S} * \mathcal{V}^T$, satisfying the tensor incoherence conditions (14–16). Let $\mathcal{R} = \mathcal{W}_\mathcal{R} * \mathcal{S}_\mathcal{R} * \mathcal{V}_\mathcal{R}^T$ be a subtensor of $\mathcal{L}$ formed by uniformly and randomly selecting $|I|$ horizontal slices of $\mathcal{L}$ such that $\mathtt{rank}_T(\mathcal{R}) = \mathtt{rank}_T(\mathcal{L})$, and $\alpha = \|\mathcal{W}_{I::}^\dagger\|$. Then,

$$\max_i \|\mathcal{W}_\mathcal{R}^T * \overrightarrow{e_i}\| \le \alpha\kappa \sqrt{\frac{\mu_1 r}{n_2 n_3}} \tag{22}$$

$$\max_j \|\mathcal{V}_\mathcal{R}^T * \overrightarrow{e_j}\| \le \sqrt{\frac{\mu_2 r}{n_1 n_3}} \tag{23}$$

$$\|\mathcal{W}_\mathcal{R} * \mathcal{V}_\mathcal{R}^T\|_\infty \le \alpha\kappa \sqrt{\frac{\mu_1 \mu_2 r}{n_1 n_2 n_3^2}} \tag{24}$$

Lemma 4: Let $\mathcal{L} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ satisfy the tensor incoherence conditions (14–16). Let $\mathcal{R} = \mathcal{L}_{I::}$ be any random subset of the lateral slices of $\mathcal{L}$ such that $\mathtt{rank}_T(\mathcal{R}) = \mathtt{rank}_T(\mathcal{L})$. Then,

$$\max_i \|\mathcal{W}_\mathcal{R}^T * \overrightarrow{e_i}\| \le \alpha\kappa \sqrt{\frac{\mu_1 r}{n_2 n_3}}$$

Proof: We have $\mathcal{R} = \mathcal{W}_\mathcal{R} * \mathcal{S}_\mathcal{R} * \mathcal{V}_\mathcal{R}^T$, and thus $\mathcal{W}_\mathcal{R}^T = \mathcal{S}_\mathcal{R}^{-T} * \mathcal{V}_\mathcal{R}^T * \mathcal{R}^T$.

$$\begin{aligned}
\max_i \|\mathcal{W}_\mathcal{R}^T * \overrightarrow{e_i}\| &= \max_i \|\mathcal{S}_\mathcal{R}^{-T} * \mathcal{V}_\mathcal{R}^T * \mathcal{R}^T * \overrightarrow{e_i}\| \\
&= \max_i \|\bar{S}_\mathcal{R}^{-H} \cdot \bar{V}_\mathcal{R}^H \cdot \bar{R}^H \cdot \overrightarrow{e_i}\| \\
&\le \|\bar{S}_\mathcal{R}^{-H}\| \|\bar{V}_\mathcal{R}^H\| \|\bar{R}^H \cdot \overrightarrow{e_i}\| \\
&= \|\bar{R}^\dagger\| \|\bar{R}^H \cdot \overrightarrow{e_i}\| \\
&= \|\bar{V}^\dagger \cdot \bar{S}^{-1} \cdot \bar{W}_{I::}^\dagger\| \|\bar{V} \cdot \bar{S}^H \cdot \bar{W}_{I::}^H \cdot \overrightarrow{e_i}\| \\
&\le \|\bar{V}^\dagger\| \|\bar{S}^{-1}\| \|\bar{W}_{I::}^\dagger\| \|\bar{V}\| \|\bar{S}^H\| \|\bar{W}_{I::}^H \cdot \overrightarrow{e_i}\| \\
&= \alpha \|\bar{S}^{-1}\| \|\bar{S}^H\| \|\mathcal{W}_{I::}^T * \overrightarrow{e_i}\| \\
&= \alpha \|\bar{L}^\dagger\| \|\bar{L}\| \|\mathcal{W}_{I::}^T * \overrightarrow{e_i}\| \\
&= \alpha\kappa \|\mathcal{W}_{I::}^T * \overrightarrow{e_i}\| \le \alpha\kappa \|\mathcal{W}^T * \overrightarrow{e_i}\|
\end{aligned}$$

Thus,

$$\max_i \|\mathcal{W}_\mathcal{R}^T * \overrightarrow{e_i}\| \le \alpha\kappa \sqrt{\frac{\mu_1 r}{n_2 n_3}}$$

□

Lemma 5: Let $\mathcal{L} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ satisfy the tensor incoherence conditions (14–16). Let $\mathcal{R} = \mathcal{L}_{I::}$ be any random subset

of the lateral slices of $\mathcal{L}$ such that $\texttt{rank}_T(\mathcal{R}) = \texttt{rank}_T(\mathcal{L})$. *Then,*

$$\max_j \|\mathcal{V}_{\mathcal{R}}^T * \vec{e_j}\| \leq \sqrt{\frac{\mu_2 r}{n_1 n_3}}$$

Proof: Note that $\mathcal{R} = \mathcal{L}_{I::} = \mathcal{W}_{I::} * \mathcal{S} * \mathcal{V}^T$. The T-SVD of $\mathcal{W}_{I::} * \mathcal{S}$ is $\mathcal{W}_{\mathcal{R}} * \mathcal{S}_{\mathcal{R}} * \hat{\mathcal{V}}_{\mathcal{C}}^T$. Thus, $\mathcal{R} = \hat{\mathcal{W}}_{\mathcal{R}} * \mathcal{S}_{\mathcal{R}} * \hat{\mathcal{V}}_{\mathcal{R}}^T * \mathcal{V}_{\mathcal{R}}^T$, and $\mathcal{V}_{\mathcal{R}} = \mathcal{V} * \hat{\mathcal{V}}_{\mathcal{R}}$. Now,

$$\begin{aligned}
\max_j \|\mathcal{V}_{\mathcal{R}}^T * \vec{e_j}\| &= \max_j \|\hat{\mathcal{V}}_{\mathcal{R}}^T * \mathcal{V}^T * \vec{e_j}\| \\
&= \max_j \|\hat{\bar{V}}_{\mathcal{R}}^H \cdot \bar{V}^H \cdot \vec{e_j}\| = \max_j \|\bar{V}^H \cdot \vec{e_j}\| \\
&= \max_j \|\mathcal{V}^T * \vec{e_j}\| \\
\max_j \|\mathcal{V}_{\mathcal{R}}^T * \vec{e_j}\| &\leq \sqrt{\frac{\mu_2 r}{n_1 n_3}}
\end{aligned}$$

□

Lemma 6: Let $\mathcal{L} = \mathcal{W} * \mathcal{S} * \mathcal{V}^T$ be a $n_1 \times n_2 \times n_3$ with tubal-rank r, satisfying the tensor incoherence conditions (14 − 16). Let $\mathcal{R} = \mathcal{L}_{I::}$ be any random subset of the lateral slices of $\mathcal{L}$ such that $\texttt{rank}_T(\mathcal{R}) = \texttt{rank}_T(\mathcal{L})$. *Then,*

$$\|\mathcal{W}_{\mathcal{R}} * \mathcal{V}_{\mathcal{R}}^T\|_\infty \leq \alpha\kappa \sqrt{\frac{\mu_1 \mu_2 r}{n_1 n_2 n_3^2}}$$

Proof: According to Lemma 4 and 5, we have $\mu_{\mathcal{C}_1} \leq \alpha^2 \kappa^2 \mu_1$ and $\mu_{\mathcal{C}_2} \leq \mu_2$. The proof follows

$$\|\mathcal{W}_{\mathcal{R}} * \mathcal{V}_{\mathcal{R}}^T\|_\infty = \|\bar{W}_{\mathcal{R}} \cdot \bar{V}_{\mathcal{R}}^H\|_\infty \leq \beta\kappa \sqrt{\frac{\mu_1 \mu_2 r}{n_1 n_2 n_3^2}}$$

□

Theorem 4: Let $\mathcal{L} = \mathcal{W} * \mathcal{S} * \mathcal{V}^T$ be a $n_1 \times n_2 \times n_3$ with tubal-rank r, satisfying the tensor incoherence conditions (14 − 16). For a positive parameter γ_2, select $|I| = \gamma_1 \mu_1 r$ horizontal slices of $\mathcal{L}$ uniformly and randomly without replacement to form subtensor $\mathcal{R} = \mathcal{L}_{I::} = \mathcal{W}_{\mathcal{I}::} * \mathcal{S} * \mathcal{V}^T$. Then, $\alpha = \|\mathcal{W}_{I::}^\dagger\|$ satisfies eq (22) and the following condition:

$$\alpha \leq \sqrt{\frac{n_1}{(1-\delta)|I|}} \tag{25}$$

with probability of at least $1 - rn_3 \cdot \left(\frac{e^{-\delta}}{(1-\delta)^{(1-\delta)}}\right)^{\gamma_2 r n_3}$ *for all* $\delta \in [0, 1)$ *[64].*

Theorem 5: Let $\mathcal{L} = \mathcal{W} * \mathcal{S} * \mathcal{V}^T$ be a $n_1 \times n_2 \times n_3$ tensor with tubal-rank r, satisfying the tensor incoherence conditions (14–16). For positive parameters γ_1, and γ_2, select $|I| = \gamma_1 \mu_1 r$ horizontal slices and $|J| = \gamma_1 \mu_2 r$ lateral slices of $\mathcal{L}$ uniformly and randomly without replacement to form subtensor $\mathcal{U} = \mathcal{L}_{IJ:}$. The following condition:

$$\|\mathcal{U}^\dagger\| \leq \frac{1}{(1-\delta)\sigma_{min}(\bar{L})} \sqrt{\frac{n_1 n_2}{|I||J|}} \tag{26}$$

satisfies with probability of at least $1 - rn_3 \cdot \left(\frac{e^{-\delta}}{(1-\delta)^{(1-\delta)}}\right)^{\gamma_1 r n_3} - rn_3 \cdot \left(\frac{e^{-\delta}}{(1-\delta)^{(1-\delta)}}\right)^{\gamma_2 r n_3}$.

Proof: We have $\mathcal{U} = \mathcal{W}_{I::} * \mathcal{S} * \mathcal{V}_{J::}^T$, thus $\mathcal{U}^\dagger = \mathcal{V}_{J::}^\dagger * \mathcal{S}^{-1} * \mathcal{W}_{I::}^\dagger$.

$$\begin{aligned}
\|\mathcal{U}^\dagger\| &= \|\mathcal{V}_{J::}^\dagger * \mathcal{S}^{-1} * \mathcal{W}_{I::}^\dagger\| \\
\|\mathcal{U}^\dagger\| &= \|\bar{V}_{J::}^\dagger \cdot \bar{S}^{-1} \cdot \bar{W}_{I::}^\dagger\| \\
\|\mathcal{U}^\dagger\| &\leq \|\bar{V}_{J::}^\dagger\| \|\bar{S}^{-1}\| \|\bar{W}_{I::}^\dagger\| \\
\|\mathcal{U}^\dagger\| &\leq \sqrt{\frac{n_2}{(1-\delta)|J|}} \frac{1}{\sigma_{min}(\bar{L})} \sqrt{\frac{n_1}{(1-\delta)|I|}} \\
\|\mathcal{U}^\dagger\| &\leq \frac{1}{(1-\delta)\sigma_{min}(\bar{L})} \sqrt{\frac{n_1 n_2}{|I||J|}}
\end{aligned}$$

□

V. RESULTS

The high dimensionality and size of hyperspectral data present computational challenges, particularly in terms of storage and processing power. Random access memory (RAM) is often a limited resource in computing systems. Increasing RAM can be costly and is subject to limitations imposed by the computer's architecture. When the input data exceeds the available memory, processing times can be substantially slowed down. Methods such as TRPCA [40], randomized TRPCA [49], RPCA [48] and PCA require the entire dataset to be stored in memory, thereby lacking the ability to reduce the memory footprint during computations. As a result, these methods possess a compression ratio of 1. In contrast, our tensor robust CUR method significantly reduces the memory footprint by heavily downsampling the input data tensor, forming small subtensors, and performing TRPCA on these subtensors. Thus, our method achieves high compression, which is essential for efficiently handling large-scale datasets.

To illustrate the effectiveness and substantial speed-up, we present the performance of our algorithm using three hyperspectral data sets namely 'Salinas,' 'Pavia University,' and 'Kennedy Space Center (KSC)'.[1] We also compared our method's performance with existing low-rank recovery methods namely TRPCA [40], randomized TRPCA [49], RPCA [48] and iterated robust CUR (IRCUR) [46] on vectorized data. For IRCUR [46], we used the default settings. After applying the aforementioned methods to the hyperspectral images, each data set is randomly divided into a 10/90 training and testing; and, support vector machine (SVM) is utilized for classification. We also applied the SVM classification to the original data sets to use them as a benchmark. All experiments are run on MATLAB 2022a using a laptop with Intel i7-11800H 2.30 GHz and 32 GB RAM.

A. SALINAS

The Salinas dataset was acquired over the Salinas Valley, California, using the NASA AVIRIS sensor, which captures

[1] https://www.ehu.eus/ccwintco/index.php?title=Hyperspectral_Remote_Sensing_Scenes

TABLE 1. Performance comparison of classification accuracy (%) for different methods on Salinas data set.

Class	Name	Train	Test	Ours10[a]	Ours15[b]	Ours20[c]	TRPCA [40]	R-TRPCA [49]	RPCA [48]	IRCUR [46]	PCA	SVM
1	Broccoli Green Weeds 1	184	1825	99.84	99.95	99.29	99.51	99.78	99.84	99.07	98.96	99.67
2	Broccoli Green Weeds 2	381	3345	100	100	99.97	100	100	99.73	99.73	99.55	99.94
3	Fallow	191	1785	100	100	100	100	100	99.89	98.71	99.61	99.78
4	Fallow Rough Plow	146	1248	99.52	98.72	98.8	99.60	99.68	99.12	98.96	98.56	99.36
5	Fallow Smooth	273	2405	99.75	99.75	99.96	99.83	99.88	99.46	98.46	99.09	99.21
6	Stubble	382	3577	100	100	100	100	100	99.86	99.97	99.78	99.86
7	Celery	401	3178	99.97	99.94	99.97	100	100	100	99.81	100	99.97
8	Grapes Untrained	1081	10190	97.83	98.53	98.88	98.04	98.30	90.43	89.95	88.66	88.58
9	Soil Vinyard Develop	608	5595	100	100	99.89	100	100	99.95	99.71	99.95	99.84
10	Corn Senesced Green Weeds	321	2957	100	99.97	100	99.93	99.53	98.38	95.74	97.6	97.97
11	Lettuce Romaine 4wk	96	972	99.9	100	100	99.9	100	97.02	95.58	93.72	97.53
12	Lettuce Romaine 5wk	199	1728	100	100	100	100	100	100	99.83	100	100
13	Lettuce Romaine 6wk	117	799	100	100	99.75	100	100	99.87	99.12	99.37	99.12
14	Lettuce Romaine 7wk	120	950	100	99.89	100	99.58	99.68	98.95	98.21	95.26	97.58
15	Vinyard Untrained	744	6524	96.29	97.18	98.16	97.39	98.04	63.53	57.11	67.49	66.71
16	Vinyard Vertical Trellis	178	1629	100	100	100	99.94	99.82	99.45	99.2	99.14	99.32
	Overall Accuracy			99.01	99.26	99.44	99.19	99.32	92.82	91.48	92.68	92.78
	Compression ratio			4.65	3.08	2.26	1	1	1	2.43	1	1
	Elapsed time (sec)			20	34	48	289	80	66	733	6	11

[a]Our method using random sampling of 10% of the lateral and 10% of the horizontal slices of the input hypercube.
[b]Our method using random sampling of 15% of the lateral and 15% of the horizontal slices of the input hypercube.
[c]Our method using random sampling of 20% of the lateral and 20% of the horizontal slices of the input hypercube.

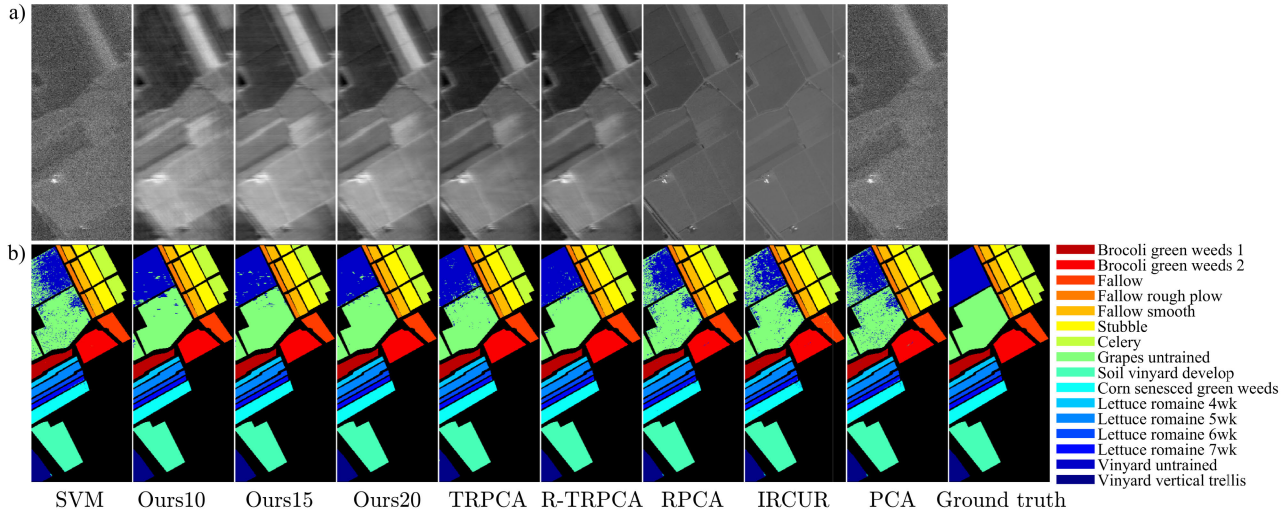

FIGURE 1. (a) displays a noisy spectral band (band 1) alongside the corresponding denoised signals for the Salinas dataset. Our method was performed using random sampling of 10%, 15%, and 20% of the horizontal and lateral slices of the input hypercube which are labeled as '*Ours*10', '*Ours*15', and '*Ours*20', respectively. The figure also shows the corresponding denoised signals obtained by TRPCA [40], R-TRPCA [49], RPCA [48], IRCUR [46], and PCA. (b) depicts the classification maps for different methods.

data in the visible and near-infrared spectrum. The dataset consists of a hyperspectral image with a spatial resolution of 3.7 meters. It contains 16 land cover classes, including vegetable cultures, vineyards, and bare soils. The dataset consists of 512 × 217 pixels and 204 spectral bands. To construct the subtensors, we randomly selected 10% of the horizontal and 10% of the lateral slices of the input hyperspectral data, resulting in $\tilde{\mathcal{R}} \in \mathbb{R}^{52\times204\times217}$ and $\tilde{\mathcal{C}} \in \mathbb{R}^{512\times21\times217}$, achieving a compression ratio of 4.65. We applied our method (Algorithm 4) to these subtensors, recovering the low-rank hyperspectral image. Our method achieved an overall classification accuracy of 99.01% on the test set. In comparison, TRPCA [40] and randomized TRPCA [49] applied to the input hypercube achieved overall classification accuracies of 99.19% and 99.32%, respectively. Notably, our tensor robust CUR method demonstrated significant speed improvements, being 14.45 times faster than TRPCA [40] and 4 times faster than randomized TRPCA [49].

To investigate the effect of the size of the subtensors on performance, two additional experiments were conducted using subtensors constructed by randomly sampling 15% and 20% of the horizontal and lateral slices of the original

TABLE 2. Performance comparison of classification accuracy (%) for different methods on Pavia University data set.

Class	Name	Train	Test	Ours10[a]	Ours15[b]	Ours20[c]	TRPCA [40]	R-TRPCA [49]	RPCA [48]	IRCUR [46]	PCA	SVM
1	Asphalt	676	5955	95.06	95.9	96.73	97.78	97.85	92.41	93.59	93.6	94.02
2	Meadows	1894	16755	98.67	99.1	99.3	99.56	99.56	96.49	97.39	97.19	97.71
3	Gravel	201	1898	79.5	89.09	92.73	88.67	90.31	70.92	55.53	72.71	75.18
4	Trees	321	2743	91.83	95.55	95.99	96.5	95.99	88.7	80.9	87.35	87.46
5	Painted metal sheets	128	1217	93.51	97.21	97.04	98.44	98.11	99.67	99.26	98.93	99.51
6	Bare Soil	486	4543	97.69	98.83	99.03	99.19	99.21	77.55	36.28	77.33	76.29
7	Bitumen	122	1208	82.86	87.67	90.89	88.58	89.82	47.85	0	50.41	52.73
8	Self-Blocking Bricks	359	3323	92.42	93.35	94.34	94.43	94.52	81.55	85.77	86.13	86.73
9	Shadows	91	856	97.66	97.2	98.48	99.42	98.48	99.42	99.77	99.88	99.65
	Overall Accuracy			95.34	96.87	97.53	97.66	97.73	89.16	82.4	90.08	90.51
	Compression ratio			4.6	3.03	2.25	1	1	1	4.31	1	1
	Elapsed time (sec)			26	39	56	96	57	61	175	3	7

[a]Our method using random sampling of 10% of the lateral and 10% of the horizontal slices of the input hypercube.
[b]Our method using random sampling of 15% of the lateral and 15% of the horizontal slices of the input hypercube.
[c]Our method using random sampling of 20% of the lateral and 20% of the horizontal slices of the input hypercube.

data tensor. These experiments resulted in slightly improved classification accuracies of 99.26% and 99.44%, respectively. However, the larger subtensor size resulted in longer processing time. Table 1 presents the classification accuracy for each method.

Matrix-based methods, such as RPCA [48] and IRCUR [46], were found to be ineffective in improving the classification accuracy. To apply IRCUR [46], we constructed submatrices $\tilde{R} \in \mathbb{R}^{20090\times 204}$ and $\tilde{C} \in \mathbb{R}^{111104\times 40}$ by randomly sampling rows and columns from the input data matrix $X \in \mathbb{R}^{111104\times 204}$.

Figure 1a provides an example of a noisy spectral band and the corresponding denoised signals obtained using different methods. Our method was performed using random sampling of 10%, 15%, and 20% of the horizontal and lateral slices of the input hypercube, which were labeled as '*Ours*10', '*Ours*15', and '*Ours*20', respectively. The figure also shows the corresponding denoised signal obtained by applying TRPCA to the entire input data tensor, serving as the reference for comparison. It is evident that our method, particularly when utilizing 20% of the horizontal and lateral slices of the input hypercube, achieves denoising result comparable to the result obtained by TRPCA directly to the original hypercube. The lower panels (Figure 1b) display the classification maps for different methods.

B. PAVIA UNIVERSITY

The Pavia University hyperspectral dataset was captured using the ROSIS sensor during a flight campaign over Pavia, northern Italy. This dataset comprises an image with dimensions of 610×340 pixels and includes 103 spectral bands that cover the wavelength range from 0.43 to 2.51 μm. The ground truth of the image provides classifications for nine distinct land cover classes, encompassing various urban materials such as asphalt, metal sheets, bricks, as well as trees.

To construct subtensors $\tilde{\mathcal{C}} \in \mathbb{R}^{610\times 11\times 340}$ and $\tilde{\mathcal{R}} \in \mathbb{R}^{61\times 103\times 340}$, we randomly selected 10% of the lateral slices and 10% of the horizontal slices from the input hypercube $\mathcal{X} \in \mathbb{R}^{610\times 103\times 340}$. Subsequently, we applied TRPCA [40] to the subtensors $\tilde{\mathcal{C}}$ and $\tilde{\mathcal{R}}$, followed by tensor CUR reconstruction to recover the low-rank hyperspectral image. Our method achieved an overall classification accuracy of 95.34%. In comparison, TRPCA [40] and randomized TRPCA [49] achieved higher overall classification accuracies of 97.66% and 97.73%, respectively. Additionally, our method demonstrated a compression ratio of 4.6 and exhibited a speed improvement of 3.7 times compared to TRPCA [40], making it 2.2 times faster than randomized TRPCA [49].

It is worth noting that increasing the sampling size to 15% and 20% of the horizontal and lateral slices of the input hyperspectral image led to improvements in classification accuracy, reaching 96.87% and 97.53%, respectively. Table 2 presents the classification accuracy for each method.

Matrix-based methods such as RPCA [48] and IRCUR [46] were found to be ineffective in improving the classification accuracy. To apply IRCUR [46], we constructed submatrices $\tilde{R} \in \mathbb{R}^{10121\times 103}$ and $\tilde{C} \in \mathbb{R}^{207400\times 18}$ by randomly sampling rows and columns from the input data matrix $X \in \mathbb{R}^{207400\times 103}$.

Figure 2a shows a noisy spectral band and the corresponding denoised signals obtained by different methods. We conducted our method by randomly selecting subsets of the horizontal and lateral slices of the input hypercube, specifically at rates of 10%, 15%, and 20%, which were labeled as '*Ours*10', '*Ours*15', and '*Ours*20', respectively. The figure also displays the denoised signal obtained by applying TRPCA to the entire input data tensor, which serves as the reference for comparison. The results indicate that our method, when utilizing 15% and 20% of the horizontal and lateral slices of the input hypercube, achieves denoising outcomes that are comparable to the result obtained by applying TRPCA directly to the original hypercube. Figure 2b

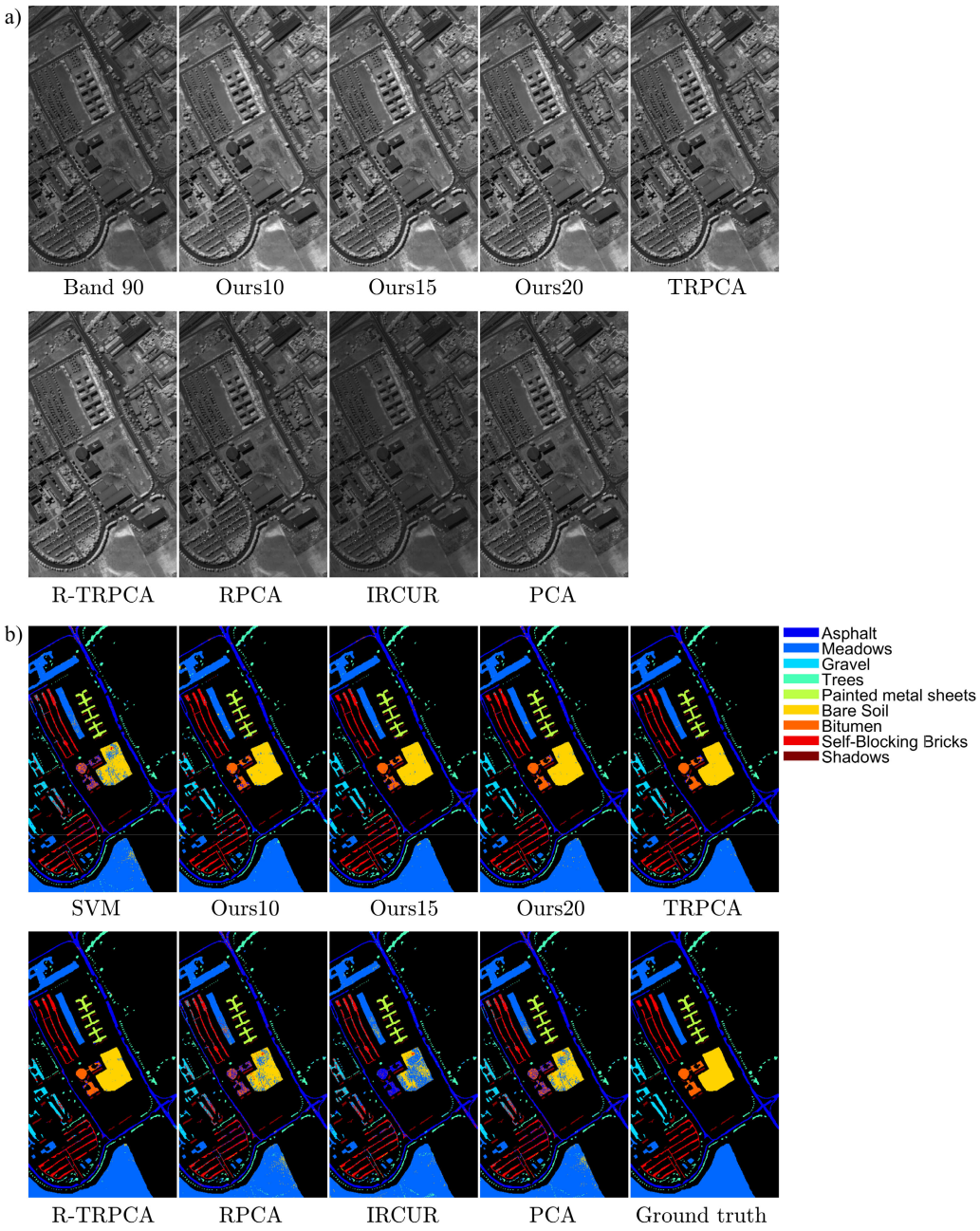

FIGURE 2. (a) displays a noisy spectral band alongside the corresponding denoised signals for the Pavia University dataset. Our method was performed using random sampling of 10%, 15%, and 20% of the horizontal and lateral slices of the input hypercube which are labeled as '*Ours*10', '*Ours*15', and '*Ours*20', respectively. The figure also shows the corresponding denoised signals obtained by TRPCA [40], R-TRPCA [49], RPCA [48], IRCUR [46], and PCA. (b) depicts the classification maps for different methods.

illustrates the classification maps obtained by each of the different methods.

C. KENNEDY SPACE CENTER

This data set was captured by the NASA AVIRIS sensor over the Kennedy Space Center (KSC). The data consists of 13 classes including various types of wild vegetation and wetlands [65]. The data is 512 × 614 pixels and 176 spectral bands. We randomly selected 10% of the lateral slices and 10% of the horizontal slices of the input hypercube $\mathcal{X} \in \mathbb{R}^{512\times176\times614}$ to construct subtensors $\tilde{\mathcal{C}} \in \mathbb{R}^{512\times18\times614}$ and $\tilde{\mathcal{R}} \in \mathbb{R}^{52\times176\times614}$. We performed our method and recovered the low-rank hyperspectral image. Our method achieved an overall classification accuracy of 99.81% which is better than the overall accuracies achieved by TRPCA [40] and randomized TRPCA [49]. The compression ratio of our method

TABLE 3. Performance comparison of classification accuracy (%) for different methods on KSC data set.

Class	Name	Train	Test	Ours10[a]	Ours15[b]	Ours20[c]	TRPCA [40]	R-TRPCA [49]	RPCA [48]	IRCUR [46]	PCA	SVM
1	Scrub	82	679	100	100	100	100	100	96.32	91.31	96.17	96.02
2	Willow	27	216	98.15	96.3	97.22	99.54	98.61	92.13	79.17	92.13	89.81
3	CP hammock	24	232	100	97.41	97.84	92.24	96.55	87.5	72.41	77.59	93.1
4	CP/oak	22	230	99.57	95.22	96.96	94.78	93.48	71.74	52.17	56.96	66.96
5	Slash pine	15	146	99.32	100	100	82.88	99.32	66.44	28.77	61.64	55.48
6	Oak/broadleaf	24	205	100	100	100	97.07	87.80	81.46	46.34	70.24	69.76
7	Hardwood swamp	14	91	100	98.9	100	97.8	100	79.12	63.74	74.73	86.81
8	Graminoid marsh	34	397	99.24	95.72	97.48	98.24	98.49	88.92	69.27	85.64	91.18
9	Spartina marsh	51	469	100	100	100	100	100	95.95	91.68	95.52	98.08
10	Cattail marsh	37	367	100	100	100	100	100	97.28	90.19	89.92	96.46
11	Salt marsh	42	377	100	98.94	100	99.73	97.88	93.9	90.19	87.53	96.82
12	Mud flats	55	448	100	100	100	99.78	99.33	99.11	84.38	96.21	98.88
13	Water	94	833	100	100	100	100	100	100	99.52	100	100
	Overall Accuracy			99.81	99.00	99.4	98.44	98.53	92.71	82.24	89.06	92.45
	Compression ratio			4.67	3.06	2.24	1	1	1	8.89	1	1
	Elapsed time (sec)			53	87	134	674	200	187	117	6	11

[a]Our method using random sampling of 10% of the lateral and 10% of the horizontal slices of the input hypercube.
[b]Our method using random sampling of 15% of the lateral and 15% of the horizontal slices of the input hypercube.
[c]Our method using random sampling of 20% of the lateral and 20% of the horizontal slices of the input hypercube.

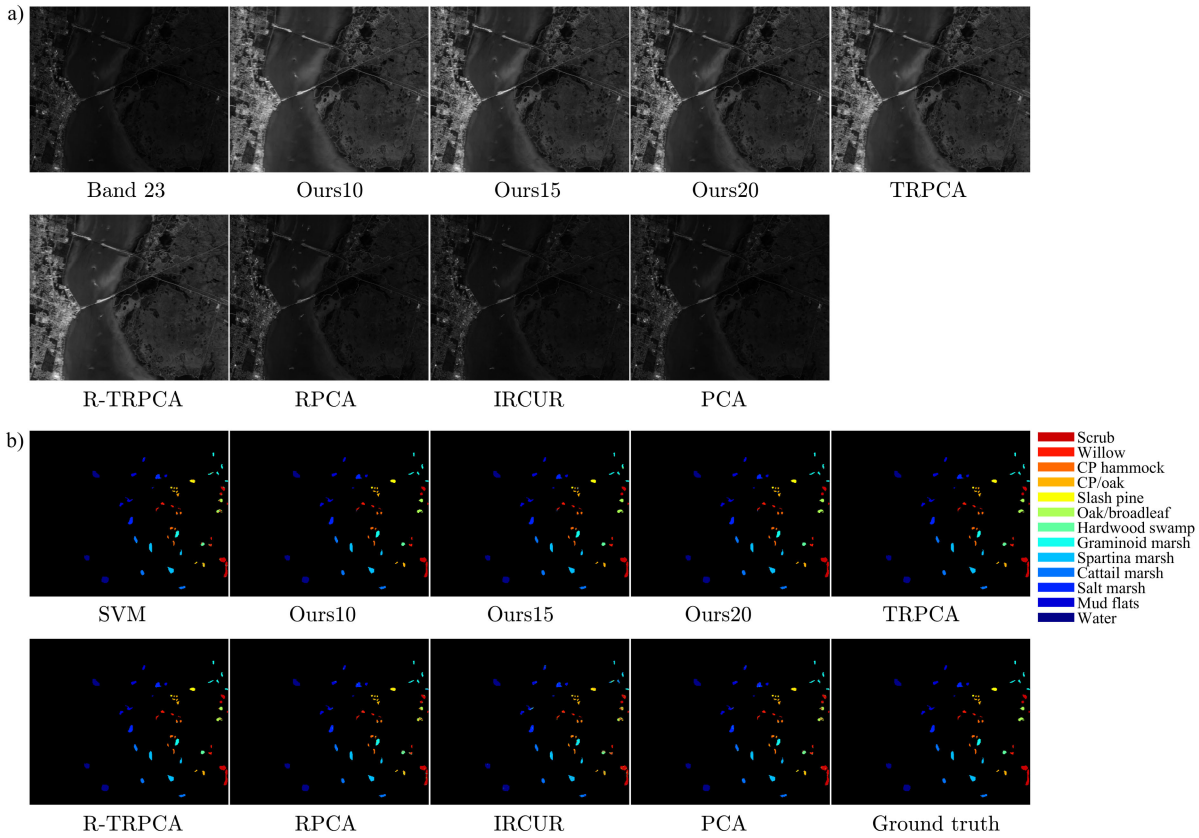

FIGURE 3. (a) displays a spectral band alongside the corresponding recovered low-rank signals for the KSC dataset. Our method was performed using random sampling of 10%, 15%, and 20% of the horizontal and lateral slices of the input hypercube which are labeled as '*Ours*10', '*Ours*15', and '*Ours*20', respectively. The figure also shows the corresponding recovered signals obtained by TRPCA [40], R-TRPCA [49], RPCA [48], IRCUR [46], and PCA. (b) depicts the classification maps for different methods.

is 4.67. Our tensor robust CUR algorithm is 12.7 times faster than TRPCA [40], and 3.8 times faster than randomized TRPCA [49]. We should also note the classification accuracy did not improve any further when the sampling size was increased to 15% and 20% of the horizontal and lateral slices of the input hyperspectral image. Table 3 shows the classification accuracy for each method.

Matrix-based methods, including RPCA [48] and IRCUR [46], were found to be ineffective in improving the classification accuracy. To apply IRCUR [46], we constructed

TABLE 4. Comparisons of average overall accuracy (%), compression ratio, and elapsed time (sec) for different methods.

Name	Ours10[a]	Ours15[b]	Ours20[c]	TRPCA [40]	R-TRPCA [49]	RPCA [48]	IRCUR [46]	PCA	SVM
Average overall accuracy (%)	98.05	98.38	98.79	98.42	98.53	91.56	85.37	90.61	91.91
Average Compression ratio	4.64	3.06	2.25	1	1	1	5.21	1	1
Elapsed time (sec)	99	160	238	1059	337	314	1025	15	29

[a]Our method using random sampling of 10% of the lateral and 10% of the horizontal slices of the input hypercube.
[b]Our method using random sampling of 15% of the lateral and 15% of the horizontal slices of the input hypercube.
[c]Our method using random sampling of 20% of the lateral and 20% of the horizontal slices of the input hypercube.

submatrices $\tilde{R} \in \mathbb{R}^{6225\times 176}$ and $\tilde{C} \in \mathbb{R}^{314368\times 16}$ by randomly sampling rows and columns from the input data matrix $X \in \mathbb{R}^{314368\times 176}$.

Figure 3a shows a spectral band and the corresponding recovered low-rank signals for different methods. Our method was performed using random sampling of 10%, 15%, and 20% of the horizontal and lateral slices of the input hypercube, which were labeled as '*Ours*10', '*Ours*15', and '*Ours*20', respectively. The results indicate that our method achieves denoising outcomes that are comparable to the result obtained by applying TRPCA directly to the original hypercube. Figure 3b illustrates the classification maps for different methods.

Table 4 displays the average overall accuracy and elapsed time for each method. The application of tensor-based methods resulted in notable improvements in classification accuracy. Our tensor robust CUR method not only achieved a significant reduction in memory usage but also accomplished the task much faster compared to other tensor-based methods [40], [49]. On average, our method, which involved random sampling of 10% of the lateral and 10% of the horizontal slices of the input hypercube, outperformed TRPCA [40] by a factor of 10.7 in terms of speed, and randomized TRPCA [49] by a factor of 3.4.

The average overall accuracy achieved by the different methods ranged from 85.37% to 98.79%. Our method exhibited a high average overall accuracy of 98.05%, which further improved to 98.38% and 98.79% when the sampling size was increased to 15% and 20%, respectively. In terms of compression ratio, our method achieved an average compression ratio of 4.64, indicating a significant reduction in the memory footprint. Comparatively, TRPCA and R-TRPCA achieved a compression ratio of 1, indicating no reduction in memory usage. In terms of elapsed time, our method exhibited remarkable efficiency, completing the analysis in just 99 seconds. This is significantly faster compared to TRPCA, which took 1059 seconds, and randomized TRPCA, which took 337 seconds. Furthermore, it is worth noting that matrix-based methods achieved lower overall accuracy when compared to tensor-based approaches. This emphasizes the advantage of utilizing tensor-based methods for hyperspectral image analysis.

VI. CONCLUSION

In this paper, we developed a tensor robust CUR method for compression and denoising of hyperspectral data. We proposed a multi-dimensional divide-and-conquer framework to drastically reduce the computational complexity of TRPCA. Numerical experiments on numerous hyperspectral data sets indicate that classification accuracy following the application of our method is as good as when performing TRPCA on the original hypercube. Results show that our algorithm is up to 14 times faster than conventional TRPCA.

REFERENCES

[1] X. Ji, Y. Cui, H. Wang, L. Teng, L. Wang, and L. Wang, "Semisupervised hyperspectral image classification using spatial–spectral information and landscape features," *IEEE Access*, vol. 7, pp. 146675–146692, 2019.

[2] C. Bo, H. Lu, and D. Wang, "Weighted generalized nearest neighbor for hyperspectral image classification," *IEEE Access*, vol. 5, pp. 1496–1509, 2017.

[3] D. Zeng, S. Zhang, F. Chen, and Y. Wang, "Multi-scale CNN based garbage detection of airborne hyperspectral data," *IEEE Access*, vol. 7, pp. 104514–104527, 2019.

[4] E. Aptoula, M. Dalla Mura, and S. Lefèvre, "Vector attribute profiles for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 54, no. 6, pp. 3208–3220, Jun. 2016.

[5] M. Khodadadzadeh, J. Li, A. Plaza, H. Ghassemian, J. M. Bioucas-Dias, and X. Li, "Spectral–spatial classification of hyperspectral data using local and global probabilities for mixed pixel characterization," *IEEE Trans. Geosci. Remote Sens.*, vol. 52, no. 10, pp. 6298–6314, Oct. 2014.

[6] X. Shang, S. Han, and M. Song, "Iterative spatial–spectral training sample augmentation for effective hyperspectral image classification," *IEEE Geosci. Remote Sens. Lett.*, vol. 19, pp. 1–5, 2022.

[7] X. Shen, W. Bao, H. Liang, X. Zhang, and X. Ma, "Grouped collaborative representation for hyperspectral image classification using a two-phase strategy," *IEEE Geosci. Remote Sens. Lett.*, vol. 19, pp. 1–5, 2022.

[8] M. Song, C. Yu, H. Xie, and C.-I. Chang, "Progressive band selection processing of hyperspectral image classification," *IEEE Geosci. Remote Sens. Lett.*, vol. 17, no. 10, pp. 1762–1766, Oct. 2020.

[9] L. Sun, B. Jeon, B. N. Soomro, Y. Zheng, Z. Wu, and L. Xiao, "Fast superpixel based subspace low rank learning method for hyperspectral denoising," *IEEE Access*, vol. 6, pp. 12031–12043, 2018.

[10] M. Jiang, F. Cao, and Y. Lu, "Extreme learning machine with enhanced composite feature for spectral–spatial hyperspectral image classification," *IEEE Access*, vol. 6, pp. 22645–22654, 2018.

[11] W. Yang, K. Hou, B. Liu, F. Yu, and L. Lin, "Two-stage clustering technique based on the neighboring union histogram for hyperspectral remote sensing images," *IEEE Access*, vol. 5, pp. 5640–5647, 2017.

[12] Z. Wang, C. Zou, and W. Cai, "Small sample classification of hyperspectral remote sensing images based on sequential joint deeping learning model," *IEEE Access*, vol. 8, pp. 71353–71363, 2020.

[13] J. Gu, T. Cheng, and B. Wang, "Reweighted kernel-based nonlinear hyperspectral unmixing with regional ℓ_1-norm regularization," *IEEE Geosci. Remote Sens. Lett.*, vol. 19, pp. 1–5, 2022.

[14] X. Wang, Y. Zhong, L. Zhang, and Y. Xu, "Spatial group sparsity regularized nonnegative matrix factorization for hyperspectral unmixing," *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 11, pp. 6287–6304, Nov. 2017.

[15] X. Tao, M. E. Paoletti, L. Han, J. M. Haut, P. Ren, J. Plaza, and A. Plaza, "Fast orthogonal projection for hyperspectral unmixing," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2022, Art. no. 5523313.

[16] S. Zhang, G. Zhang, F. Li, C. Deng, S. Wang, A. Plaza, and J. Li, "Spectral–spatial hyperspectral unmixing using nonnegative matrix factorization," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2022, Art. no. 5505713.

[17] B. Palsson, J. Sigurdsson, J. R. Sveinsson, and M. O. Ulfarsson, "Hyperspectral unmixing using a neural network autoencoder," *IEEE Access*, vol. 6, pp. 25646–25656, 2018.

[18] R. Macias, S. Bernabé, D. Báscones, and C. González, "FPGA implementation of a hardware optimized automatic target detection and classification algorithm for hyperspectral image analysis," *IEEE Geosci. Remote Sens. Lett.*, vol. 19, pp. 1–5, 2022.

[19] 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.

[20] T. Wang, H. Zhang, H. Lin, and X. Jia, "A sparse representation method for a priori target signature optimization in hyperspectral target detection," *IEEE Access*, vol. 6, pp. 3408–3424, 2018.

[21] K. L. Ang and J. K. P. Seng, "Big data and machine learning with hyperspectral information in agriculture," *IEEE Access*, vol. 9, pp. 36699–36718, 2021.

[22] M. J. Khan, H. S. Khan, A. Yousaf, K. Khurshid, and A. Abbas, "Modern trends in hyperspectral image analysis: A review," *IEEE Access*, vol. 6, pp. 14118–14129, 2018.

[23] M. D. Farrell and R. M. Mersereau, "On the impact of PCA dimension reduction for hyperspectral detection of difficult targets," *IEEE Geosci. Remote Sens. Lett.*, vol. 2, no. 2, pp. 192–195, Apr. 2005.

[24] G. Chen and S.-E. Qian, "Denoising of hyperspectral imagery using principal component analysis and wavelet shrinkage," *IEEE Trans. Geosci. Remote Sens.*, vol. 49, no. 3, pp. 973–980, Mar. 2011.

[25] Z. Khan, F. Shafait, and A. Mian, "Joint group sparse PCA for compressed hyperspectral imaging," *IEEE Trans. Image Process.*, vol. 24, no. 12, pp. 4934–4942, Dec. 2015.

[26] F. Palsson, J. R. Sveinsson, M. O. Ulfarsson, and J. A. Benediktsson, "Model-based fusion of multi- and hyperspectral images using PCA and wavelets," *IEEE Trans. Geosci. Remote Sens.*, vol. 53, no. 5, pp. 2652–2663, May 2015.

[27] M. Huber-Lerner, O. Hadar, S. R. Rotman, and R. Huber-Shalem, "Compression of hyperspectral images containing a subpixel target," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 7, no. 6, pp. 2246–2255, Jun. 2014.

[28] X. Kang, X. Xiang, S. Li, and J. A. Benediktsson, "PCA-based edge-preserving features for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 12, pp. 7140–7151, Dec. 2017.

[29] Y. Yan, J. Ren, Q. Liu, H. Zhao, H. Sun, and J. Zabalza, "PCA-domain fused singular spectral analysis for fast and noise-robust spectral–spatial feature mining in hyperspectral classification," *IEEE Geosci. Remote Sens. Lett.*, vol. 20, pp. 1–5, 2021.

[30] X. Zhang, X. Jiang, J. Jiang, Y. Zhang, X. Liu, and Z. Cai, "Spectral–spatial and superpixelwise PCA for unsupervised feature extraction of hyperspectral imagery," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2022, Art. no. 5502210.

[31] J. Jiang, J. Ma, C. Chen, Z. Wang, Z. Cai, and L. Wang, "SuperPCA: A superpixelwise PCA approach for unsupervised feature extraction of hyperspectral imagery," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 8, pp. 4581–4593, Aug. 2018.

[32] J. A. Jablonski, T. J. Bihl, and K. W. Bauer, "Principal component reconstruction error for hyperspectral anomaly detection," *IEEE Geosci. Remote Sens. Lett.*, vol. 12, no. 8, pp. 1725–1729, Aug. 2015.

[33] C. Lee, S. Youn, T. Jeong, E. Lee, and J. Serra-Sagristà, "Hybrid compression of hyperspectral images based on PCA with pre-encoding discriminant information," *IEEE Geosci. Remote Sens. Lett.*, vol. 12, no. 7, pp. 1491–1495, Jul. 2015.

[34] X. Liu, S. Bourennane, and C. Fossati, "Nonwhite noise reduction in hyperspectral images," *IEEE Geosci. Remote Sens. Lett.*, vol. 9, no. 3, pp. 368–372, May 2012.

[35] D. Letexier and S. Bourennane, "Noise removal from hyperspectral images by multidimensional filtering," *IEEE Trans. Geosci. Remote Sens.*, vol. 46, no. 7, pp. 2061–2069, Jul. 2008.

[36] N. Renard, S. Bourennane, and J. Blanc-Talon, "Denoising and dimensionality reduction using multilinear tools for hyperspectral images," *IEEE Geosci. Remote Sens. Lett.*, vol. 5, no. 2, pp. 138–142, Apr. 2008.

[37] W. Sun, G. Yang, J. Peng, and Q. Du, "Lateral-slice sparse tensor robust principal component analysis for hyperspectral image classification," *IEEE Geosci. Remote Sens. Lett.*, vol. 17, no. 1, pp. 107–111, Jan. 2020.

[38] S. Meng, L.-T. Huang, and W.-Q. Wang, "Tensor decomposition and PCA jointed algorithm for hyperspectral image denoising," *IEEE Geosci. Remote Sens. Lett.*, vol. 13, no. 7, pp. 897–901, Jul. 2016.

[39] Z. Huang, S. Li, L. Fang, H. Li, and J. A. Benediktsson, "Hyperspectral image denoising with group sparse and low-rank tensor decomposition," *IEEE Access*, vol. 6, pp. 1380–1390, 2018.

[40] C. Lu, J. Feng, Y. Chen, W. Liu, Z. Lin, and S. Yan, "Tensor robust principal component analysis with a new tensor nuclear norm," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 4, pp. 925–938, Apr. 2020.

[41] J. M. Bioucas-Dias, A. Plaza, G. Camps-Valls, P. Scheunders, N. 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.

[42] J. Lee, X. Cai, J. Lellmann, M. Dalponte, Y. Malhi, N. Butt, M. Morecroft, C.-B. Schönlieb, and D. A. Coomes, "Individual tree species classification from airborne multisensor imagery using robust PCA," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 9, no. 6, pp. 2554–2567, Jun. 2016.

[43] S. Rambhatla, X. Li, J. Ren, and J. Haupt, "A dictionary-based generalization of robust PCA with applications to target localization in hyperspectral imaging," *IEEE Trans. Signal Process.*, vol. 68, pp. 1760–1775, 2020.

[44] Y. Liu, Q. Zhang, Y. Chen, Q. Cheng, and C. Peng, "Hyperspectral image denoising with log-based robust PCA," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2021, pp. 1634–1638.

[45] B. Wu, J. Yu, H. Ren, Y. Lou, and N. Liu, "Seismic traffic noise attenuation using ℓ_p-norm robust PCA," *IEEE Geosci. Remote Sens. Lett.*, vol. 17, no. 11, pp. 1998–2001, Nov. 2019.

[46] H. Cai, K. Hamm, L. Huang, J. Li, and T. Wang, "Rapid robust principal component analysis: CUR accelerated inexact low rank estimation," *IEEE Signal Process. Lett.*, vol. 28, pp. 116–120, 2021.

[47] H. Cai, K. Hamm, L. Huang, and D. Needell, "Robust CUR decomposition: Theory and imaging applications," *SIAM J. Imag. Sci.*, vol. 14, no. 4, pp. 1472–1503, Jan. 2021.

[48] E. J. Candès, X. Li, Y. Ma, and J. Wright, "Robust principal component analysis?" *J. ACM*, vol. 58, no. 3, pp. 1–37, 2011.

[49] M. M. Salut and D. V. Anderson, "Randomized tensor robust PCA for noisy hyperspectral image classification," *IEEE Geosci. Remote Sens. Lett.*, vol. 20, pp. 1–5, 2023.

[50] M. W. Mahoney, M. Maggioni, and P. Drineas, "Tensor-CUR decompositions for tensor-based data," in *Proc. 12th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2006, pp. 327–336.

[51] Y. Yang, S. Song, D. Liu, J. C. Chan, J. Li, and J. Zhang, "Hyperspectral anomaly detection through sparse representation with tensor decomposition-based dictionary construction and adaptive weighting," *IEEE Access*, vol. 8, pp. 72121–72137, 2020.

[52] F. Ma, F. Yang, and Y. Wang, "Low-rank tensor decomposition with smooth and sparse regularization for hyperspectral and multispectral data fusion," *IEEE Access*, vol. 8, pp. 129842–129856, 2020.

[53] H. Cai, Z. Chao, L. Huang, and D. Needell, "Fast robust tensor principal component analysis via fiber CUR decomposition," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. Workshops (ICCVW)*, Oct. 2021, pp. 189–197.

[54] M. E. Kilmer, K. Braman, N. Hao, and R. C. Hoover, "Third-order tensors as operators on matrices: A theoretical and computational framework with applications in imaging," *SIAM J. Matrix Anal. Appl.*, vol. 34, no. 1, pp. 148–172, Jan. 2013.

[55] M. E. Kilmer and C. D. Martin, "Factorization strategies for third-order tensors," *Linear Algebra Appl.*, vol. 435, no. 3, pp. 641–658, Aug. 2011.

[56] M. E. Kilmer, L. Horesh, H. Avron, and E. Newman, "Tensor-tensor algebra for optimal representation and compression of multiway data," *Proc. Nat. Acad. Sci. USA*, vol. 118, no. 28, pp. 1–12, Jul. 2021.

[57] A. Zare, A. Ozdemir, M. A. Iwen, and S. Aviyente, "Extension of PCA to higher order data structures: An introduction to tensors, tensor decompositions, and tensor PCA," *Proc. IEEE*, vol. 106, no. 8, pp. 1341–1358, Aug. 2018.

[58] M. M. Salut and D. V. Anderson, "Online tensor robust principal component analysis," *IEEE Access*, vol. 10, pp. 69354–69363, 2022.

[59] L. Wang, K. Xie, T. Semong, and H. Zhou, "Missing data recovery based on tensor-CUR decomposition," *IEEE Access*, vol. 6, pp. 532–544, 2018.

[60] J. Chen, Y. Wei, and Y. Xu, "Tensor CUR decomposition under T-product and its perturbation," *Numer. Funct. Anal. Optim.*, vol. 43, no. 6, pp. 1–25, 2022.

[61] Z. Zhang and S. Aeron, "Exact tensor completion using t-SVD," *IEEE Trans. Signal Process.*, vol. 65, no. 6, pp. 1511–1526, Mar. 2017.

[62] Y. Su, X. Wu, and W. Liu, "Low-rank tensor completion by sum of tensor nuclear norm minimization," *IEEE Access*, vol. 7, pp. 134943–134953, 2019.

[63] Y. Chen, "Incoherence-optimal matrix completion," *IEEE Trans. Inf. Theory*, vol. 61, no. 5, pp. 2909–2923, May 2015.

[64] J. A. Tropp, "Improved analysis of the subsampled randomized Hadamard transform," *Adv. Adapt. Data Anal.*, vol. 3, pp. 115–126, Apr. 2011.

[65] N. Audebert, B. Le Saux, and S. Lefevre, "Deep learning for classification of hyperspectral data: A comparative review," *IEEE Geosci. Remote Sens. Mag.*, vol. 7, no. 2, pp. 159–173, Jun. 2019.

MOHAMMAD M. SALUT (Graduate Student Member, IEEE) received the M.S. degree in electrical and computer engineering from the Georgia Institute of Technology, in 2018, where he is currently pursuing the Ph.D. degree in electrical and computer engineering. His current research interests include digital signal processing, machine learning, and tensor data analysis.

DAVID V. ANDERSON (Senior Member, IEEE) received the B.S. and M.S. degrees from Brigham Young University, in 1993 and 1994, respectively, and the Ph.D. degree from the Georgia Institute of Technology, in 1999. He is currently a Professor with the School of Electrical and Computer Engineering, Georgia Tech.