Received 27 May 2022, accepted 18 June 2022, date of publication 27 June 2022, date of current version 5 July 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3186364

Online Tensor Robust Principal Component Analysis

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 Online robust principal component analysis (RPCA) algorithms recursively decompose incoming data into low-rank and sparse components. However, they operate on data vectors and cannot directly be applied to higher-order data arrays (e.g. video frames). In this paper, we propose a new online robust PCA algorithm that preserves the multi-dimensional structure of data. Our algorithm is based on the recently proposed tensor singular value decomposition (T-SVD). We develop a convex optimization-based approach to recover the sparse component; and subsequently, update the low-rank component using incremental T-SVD. We propose an efficient tensor convolutional extension to the fast iterative shrinkage thresholding algorithm (FISTA) to produce a fast algorithm to solve this optimization problem. We demonstrate tensor-RPCA with the application of background foreground separation in a video stream. The foreground component is modeled as a sparse signal. The background component is modeled as a gradually changing low-rank subspace. Extensive experiments on real-world videos are presented and results demonstrate the effectiveness of our online tensor robust PCA.

INDEX TERMS Multilinear subspace learning, tensor convolutional sparse coding, low-rank tensor model, tensor singular value decomposition (T-SVD).

I. INTRODUCTION

Robust principal component analysis (RPCA) decomposes a given data matrix into a low-rank matrix and a sparse matrix. RPCA has been widely used in many computer vision applications, including video surveillance [1], background substitution [2], video coding [3], and action recognition [4]. Conventional RPCA requires batch processing [5]–[9]. Given a video sequence, all frames are reshaped into very long vectors to form the columns of a data matrix. Then, the data matrix is stored in memory for the SVD computation. Most RPCA algorithms require many iterations to converge [10]. As a result, batch-based methods require high memory overhead and are computationally prohibitive for large-scale datasets.

In many real-world applications (e.g. traffic density estimation [11]) the data frames are initially unavailable and come in sequentially. In these applications, we are interested in estimating the low-rank and sparse components on the fly or with only some small delays. Several recursive RPCA algorithms have been developed to solve the memory and computational bottleneck (e.g., [12]–[15], [15]–[21]). These algorithms process the incoming data frames recursively and then discard them. Therefore, they are usually memory-efficient and fast. However, all of the cited methods can only process 1-way (vector) data and cannot directly be applied to higher-order data arrays.

Often data is generated as higher-order tensors (e.g. video frames), possessing intrinsic structure amongst multiple elements at once [22]–[25]. The vectorization of multidimensional signals can break the intrinsic structure of data. Therefore, a higher-order extension of RPCA for multidimensional signals becomes highly desirable. Recently, RPCA has been extended from 2D matrices to 3-way tensors to capture multi-factor structure in data. It was shown that the resulting tensor robust PCA using tensor nuclear norm can exactly recover the low-rank and sparse tensors from their sum [26].

Tensor robust PCA has been used for background subtraction and foreground detection. The background contains stationary objects and undergoes gradual changes over

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

time, and so is well modeled as a low-rank tensor. The foreground captures moving objects/regions and is modeled as a sparse tensor. Results from [26] indicate that tensor RPCA outperforms conventional RPCA. However, batch tensor RPCA is associated with a long run time.

We propose an online tensor robust PCA algorithm that preserves the multi-dimensional structures of data. Big– or streaming–data processing can be significantly improved by recursive approaches to solve tensor RPCA. At time instance t, we wish to decompose a video frame ($\overrightarrow{\mathcal{X}}_t \in \mathbb{R}^{n_1 \times 1 \times n_3}$) into low-rank background ($\overrightarrow{\mathcal{L}}_t$) and sparse foreground ($\overrightarrow{\mathcal{E}}_t$) components. Our algorithm first recovers the sparse component ($\overrightarrow{\mathcal{E}}_t$) using a tensor convolutional basis pursuit framework. Second, the low-rank component is recovered by subtracting the sparse component from the original data frame ($\overrightarrow{\mathcal{L}}_t = \overrightarrow{\mathcal{X}}_t - \overrightarrow{\mathcal{E}}_t$). Then, the low-rank subspace is updated to track/learn the subspace changes. The main contributions of this work are as follows.

1) We develop a fast and memory-efficient tensor RPCA algorithm for real-time processing of streaming data.
2) We propose an efficient tensor convolutional extension to the fast iterative shrinkage thresholding algorithm (FISTA) to solve the sparse coding problem.
3) We develop an incremental tensor SVD method to track the subspace changes.
4) We evaluate the performance of our algorithm using real-world videos. Results show that our online tensor robust PCA method generally outperforms other background and foreground separation methods for video streams.

II. RELATED WORK

Robust PCA algorithms have been widely used in signal processing applications such as video surveillance, inpainting, behavior recognition, motion segmentation. Batch-based RPCA algorithms [5]–[8], [10], [26]–[33]; despite their great performance, require large memory and their complexity grows with batch size. Moreover, batch methods in which all data needs to be accessed prior to commencement of the processing are not suited for streaming environments. To address these issues, incremental RPCA methods have been developed to significantly reduce the memory and computational cost. Some recent successful variations include [13], [16], and [34]–[36]. He *et al.* [43] used an l_1-norm loss function and developed a Grassmannian robust subspace tracking method (GRASTA) for high quality background and foreground separation. Seidel *et al.*. [44] replaced the ℓ_1-norm with a smoothed ℓ_p-quasi-norm ($p < 1$) to develop an online background subtraction algorithm. Experimental results indicate that their method [44] outperforms GRASTA. Rodriguez and Wohlberg [45] proposed an incremental PCP algorithm for foreground detection. Their algorithm [45] is able to handle jitter or camera motion. Javed *et al.*. [46] developed a superpixel based matrix decomposition method for foreground detection in real time. Narayanamurthy *et al.*. [13] developed a subspace tracking method in the presence of missing data based on recursive projected compressive sensing. Comprehensive reviews on online robust PCA and background/foreground separation can be found in [20], [21], and [35]–[40]. However, all the aforementioned methods can only process 1-way (vector) data.

Many real world signals such as images and video frames possess multidimensional structures. As previously noted, RPCA algorithms can only process data in the form of vectors. Therefore, as a preprocessing stage, higher order tensors need to be converted to very long vectors, breaking the intrinsic structure of the data. Recently, several online *tensor* RPCA algorithms have been proposed [47]–[51] to avoid this problem. Anandkumar *et al.*. [47] proposed a non-convex iterative algorithm for robust tensor CANDECOMP/PARAFAC (CP) decomposition.[1] Kasai [52], [53] proposed a subspace tracking method based on CP decomposition using second-order stochastic gradient descent. Ozdemir *et al.*. [48] developed a recursive sparse and low-rank recovery algorithm using Tucker decomposition for dynamic brain network analysis. Their method [48] has high complexity $O(\alpha N^5)$. Zhang *et al.*. [54] proposed an online tensor-based RPCA algorithm for cloud removal from satellite images. The results appeared promising on the two images that they presented; additionally, the authors show that their algorithm requires very few iterations to converge. However, each iteration requires computing a matrix inversion on the frontal slices which can be computationally prohibitive for many applications. No other quantitative performance metrics were given. Sobral *et al.*. [55] proposed an incremental subspace learning algorithm based on a multifeature tensor model and Tucker decomposition for background subtraction for video streams. While effective, the processing speed was insufficient for some applications (the authors reported a separation rate of 2 minutes per frame for frame size of 128×160 using an Intel Core i7-3740qm 2.7 GHz processor).

In this paper, we propose a memory–efficient online tensor robust PCA algorithm for background/foreground separation based on the recently proposed T-product and T-SVD factorization [56]–[58]. Our algorithm processes the incoming video frames recursively, and then discards them; hence, it is memory efficient. We develop a fast inverse-free tensor convolutional sparse coding framework based on the fast iterative shrinkage thresholding algorithm (FISTA) to detect the foreground. Our algorithm achieves a separation rate of 0.04 second per frame for frame size 128×160 on a computer using AMD Ryzen 7 2700X 3.70 GHz. We used ground-truth-based metrics to quantify the performance of our algorithm using multiple video sequences in the CDnet dataset [59].

III. PRELIMINARIES

In this paper, scalars are denoted by lowercase letters, e.g., x. Vectors are denoted by boldface lowercase letters, e.g., $\boldsymbol{x}$. Matrices are denoted by capital letters, e.g., X. Third-order

[1]The CP decomposition factorizes a tensor into a sum of rank one outer products. The Tucker decomposition decomposes a tensor into a small core tensor multiplied by a set of basis matrices.

tensors are denoted by Euler script letters, e.g., $\mathcal{X} \in \mathbb{R}^{n_1\times n_2\times n_3}$. The ith lateral slice of $\mathcal{X}$ is denoted by a tensor $\overrightarrow{\mathcal{X}_i} \equiv \mathcal{X}(:,i,:)$. The ith frontal slice of $\mathcal{X}$ is denoted by a matrix $X^{(i)} \equiv \mathcal{X}(:,:,i)$ [56]–[58]. The ith mode-1 (column) fiber of the lateral slice $\overrightarrow{\mathcal{E}} \in \mathbb{R}^{n_1\times 1\times n_3}$ is denoted by a vector $\overrightarrow{\mathcal{E}}^{(i)} \equiv \boldsymbol{e}_i$ where $\boldsymbol{e}_i \in \mathbb{R}^{n_1}$. The block circulant matrix of a tensor is define as:

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

The block circulant matrix can be diagonalized by a discrete Fourier transform (DFT):

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

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

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

We can matricize a tensor by unfolding its frontal slices.

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

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

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

T-product. Let $\mathcal{X} \in \mathbb{R}^{n_1\times n_2\times n_3}$ and $\mathcal{Y} \in \mathbb{R}^{n_2\times n_4\times n_3}$. The T-product $\mathcal{Z} \in \mathbb{R}^{n_1\times n_4\times n_3}$ is defined as [57]

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

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.

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

By multiplying both sides of (6) with $\left(F_{n_3}\otimes I_{n_1}\right)$, we obtain

$$\bar{Z}^{(i)} = \bar{X}^{(i)}\cdot\bar{Y}^{(i)} \tag{8}$$

Algorithm 1 T-Product

```
Input: $\mathcal{X} \in \mathbb{R}^{n_1\times n_2\times n_3}$, $\mathcal{Y} \in \mathbb{R}^{n_2\times n_4\times n_3}$
Output: $\mathcal{Z} \in \mathbb{R}^{n_1\times n_4\times n_3}$
1: $\bar{\mathcal{X}} = \texttt{fft}(\mathcal{X},[\,],3)$;   $\bar{\mathcal{Y}} = \texttt{fft}(\mathcal{Y},[\,],3)$
2: for $i = 1 : \lceil\frac{n_3+1}{2}\rceil$ do
3:     $\bar{Z}^{(i)} = \bar{X}^{(i)}\,\bar{Y}^{(i)}$
4: end for
5: for $i = \lceil\frac{n_3+1}{2}\rceil + 1 : n_3$ do
6:     $\bar{Z}^{(i)} = \texttt{conj}\left(\bar{Z}^{(n_3-i+2)}\right)$
7: end for
8: $\mathcal{Z} = \texttt{ifft}(\bar{\mathcal{Z}},[\,],3)$
```

which is equivalent to multiplying each frontal slice of $\bar{\mathcal{X}}$ with its counterpart in $\bar{\mathcal{Y}}$ [26]. Due to the conjugate symmetry property of the DFT for real-valued data, we only need to compute $\bar{Z}^{(i)} = \bar{X}^{(i)}\,\bar{Y}^{(i)}$ for about half of the transformed frontal slices. The T-product is summarized in Algorithm 1.

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

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

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 $\bar{\mathcal{I}}^{(i)} = I$.
Orthogonal tensor. A tensor $\mathcal{U} \in \mathbb{R}^{n_1\times n_1\times n_3}$ is orthogonal if $\mathcal{U} * \mathcal{U}^T = \mathcal{U}^T * \mathcal{U} = \mathcal{I}$. Where $(*)$ represents T-product.

F-diagonal tensor. A tensor is f-diagonal if each frontal slice is a diagonal matrix.

T-SVD. T-SVD of a tensor $\mathcal{X} \in \mathbb{R}^{n_1\times n_2\times n_3}$ is the factorization of $\mathcal{X}$ into the T-product of three tensors $\mathcal{X} = \mathcal{U} * \mathcal{S} * \mathcal{V}^T$. Where $\mathcal{U} \in \mathbb{R}^{n_1\times n_1\times n_3}$ 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. Algorithm 2 summarizes T-SVD factorization for real-valued data.

The Frobenius norm of a tensor is given by:

$$\parallel \mathcal{X} \parallel_F = \sqrt{\sum_{i=1}^{n_1}\sum_{j=1}^{n_2}\sum_{k=1}^{n_3} x_{ijk}^2}$$

The ℓ_1-norm of a tensor is given by: $\parallel \mathcal{X} \parallel_1 = \sum_{ijk} \mid x_{ijk} \mid$.

IV. ONLINE TENSOR ROBUST PCA

We formulate background/foreground separation in a video stream as a recursive tensor robust PCA. The video background changes slowly over time; and is approximated as a low-rank tensor. Whereas, the foreground part changes faster than the background and contains the moving objects/regions. The foreground component is modeled as a sparse tensor.

$$\overrightarrow{\mathcal{X}}_t = \overrightarrow{\mathcal{L}}_t + \overrightarrow{\mathcal{E}}_t \tag{9}$$

Algorithm 2 T-SVD

```
Input: $\mathcal{X}$
Output: $\mathcal{U}$, $\mathcal{S}$, $\mathcal{V}$
1: $\bar{\mathcal{X}} = \texttt{fft}(\mathcal{X}, [\,], 3)$;
2: for $i = 1 : \lceil \frac{n_3+1}{2} \rceil$ do
3:     $[\bar{U}^{(i)} \quad \bar{S}^{(i)} \quad \bar{V}^{(i)}] = \texttt{svd}\left(\bar{X}^{(i)}\right)$
4: end for
5: for $i = \lceil \frac{n_3+1}{2} \rceil + 1 : n_3$ do
6:     $\bar{U}^{(i)} = \texttt{conj}\left(\bar{U}^{(n_3-i+2)}\right)$
7:     $\bar{S}^{(i)} = \bar{S}^{(n_3-i+2)}$
8:     $\bar{V}^{(i)} = \texttt{conj}\left(\bar{V}^{(n_3-i+2)}\right)$
9: end for
10: $\mathcal{U} = \texttt{ifft}(\bar{\mathcal{U}}, [\,], 3)$
11: $\mathcal{S} = \texttt{ifft}(\bar{\mathcal{S}}, [\,], 3)$
12: $\mathcal{V} = \texttt{ifft}(\bar{\mathcal{V}}, [\,], 3)$
```

where $\overrightarrow{\mathcal{X}}_t$, $\overrightarrow{\mathcal{L}}_t$ and $\overrightarrow{\mathcal{E}}_t \in \mathbb{R}^{n_1 \times 1 \times n_3}$ are a given video frame, low-rank background, and sparse foreground at time instance t, respectively. Our recursive algorithm consists of three stages: sparse recovery, low-rank recovery, and subspace tracking stages.

The algorithm starts ($t = 0$) by finding an initial estimate of the subspace of the low-rank background using training data by performing tensor singular value thresholding (T-SVT) on training data which is outlier free. This generates $\mathcal{U}_0 \in \mathbb{R}^{n_1 \times d \times n_3}$ where $d < min(n_1, n_2)$. n_2 denotes the number of video frames. The T-SVT is computed as [26]

$$\mathcal{T}_\theta(\mathcal{X}_0) = \mathcal{U}_0 * \mathcal{T}_\theta(\mathcal{S}_0) * \mathcal{V}_0^T \tag{10}$$

where $\mathcal{T}$ is the soft-thresholding operator defined as: $\mathcal{T}_\theta(\mathcal{S}_0) = \texttt{ifft}((\bar{\mathcal{S}}_0 - \theta)_+, [\,], 3)$ for any $\theta > 0$. In the absence of outlier free training data, the initial subspace of the low-rank background can be estimated by performing batch tensor RPCA on the training data [26]. We formulate foreground detection as a sparse coding problem.

$$\overrightarrow{\mathcal{X}}_t = \overrightarrow{\mathcal{L}}_t + \mathcal{D}_t * \overrightarrow{\mathcal{Z}}_t \tag{11}$$

Given a video frame $\overrightarrow{\mathcal{X}_t}$, we can recover sparse coefficients $\overrightarrow{\mathcal{Z}_t}$ if the dictionary tensor $\mathcal{D}_t$ is known. We obtain the foreground dictionary ($\mathcal{D}_t$) as the subspace orthogonal to the subspace of the low-rank background.

$$\mathcal{D}_t = \mathcal{I} - \mathcal{U}_{t-1} * \mathcal{U}_{t-1}^T \tag{12}$$

This particular choice of dictionary will be shown later to provide significant computational advantages. Since the subspace of the sparse foreground resides outside the subspace of the low-rank background, the dictionary only recovers the foreground component ($\overrightarrow{\mathcal{E}}_t$) from the video frame. The sparse foreground component is expressed as T-linear combination $\left(\overrightarrow{\mathcal{E}}_t \approx \mathcal{D}_t * \overrightarrow{\mathcal{Z}}_t\right)$ of the tensor dictionary slices $\mathcal{D}_t = \{\overrightarrow{\mathcal{D}}_1, \ldots, \overrightarrow{\mathcal{D}}_{n1}\}$ where $\overrightarrow{\mathcal{Z}}_t$ is a sparse tensor representing the dictionary coefficients. The corresponding basis pursuit problem is defined as

$$\underset{\overrightarrow{\mathcal{Z}}_t}{\arg\min} \quad \frac{1}{2}\|\mathcal{D}_t * \overrightarrow{\mathcal{Z}}_t - \overrightarrow{\mathcal{X}}_t\|_2^2 + \lambda\|\overrightarrow{\mathcal{Z}}_t\|_1 \tag{13}$$

where $f(\overrightarrow{\mathcal{Z}}_t) = \frac{1}{2}\|\mathcal{D}_t * \overrightarrow{\mathcal{Z}}_t - \overrightarrow{\mathcal{X}}_t\|_2^2$ is a smooth function and continuously differentiable with the Lipschitz continuous gradient $L(f)$, satisfying $\|\nabla f(\boldsymbol{z}) - \nabla f(\boldsymbol{y})\| \leq L(f)\|\boldsymbol{z} - \boldsymbol{y}\|$ for every $\boldsymbol{z}, \boldsymbol{y} \in \mathbb{R}^n$ where ∇ is the gradient operator. $g(\overrightarrow{\mathcal{Z}}_t) = \lambda\|\overrightarrow{\mathcal{Z}}_t\|_1$ is non-smooth ℓ_1 regularization term; and λ is the regularization parameter which controls the sparsity. A reasonable value for the regularization parameter is $\lambda = \frac{1}{\sqrt{n_1 n_3}}$ but it can be adjusted by the user to further improve the performance [5]. We employ the fast iterative shrinkage-thresholding algorithm (FISTA) [60]–[62] to minimize the objective function. The FISTA method optimizes $f(\overrightarrow{\mathcal{Z}}_t)$ through an iterative scheme given by:

$$\begin{aligned}\overrightarrow{\mathcal{Z}}_{t(j)} = \text{argmin}\Big\{ & f(\overrightarrow{\mathcal{Z}}_{t(j-1)}) \\ & + \left\langle \overrightarrow{\mathcal{Z}}_t - \overrightarrow{\mathcal{Z}}_{t(j-1)}, \nabla f(\overrightarrow{\mathcal{Z}}_{t(j-1)}) \right\rangle \\ & + \frac{L(f)}{2}\|\overrightarrow{\mathcal{Z}}_t - \overrightarrow{\mathcal{Z}}_{t(j-1)}\|^2 + \lambda\|\overrightarrow{\mathcal{Z}}_t\|_1 \Big\}\end{aligned} \tag{14}$$

where sub-indices $(j-1)$ and (j) denote iteration numbers and $\langle\cdot,\cdot\rangle$ is the inner product operator. $\overrightarrow{\mathcal{Z}}_{t(j)}$ has a closed-form solution in the form of a soft thresholding function as follows:

$$\overrightarrow{\mathcal{Z}}_{t_{(j)}} = \mathcal{S}_{\frac{\lambda}{L}}\left(\overrightarrow{\mathcal{Z}}_{t_{(j-1)}} - \frac{1}{L}\nabla\left(\frac{1}{2}\|\mathcal{D}_t * \overrightarrow{\mathcal{Z}}_{t_{(j-1)}} - \overrightarrow{\mathcal{X}}_t\|_2^2\right)\right) \tag{15}$$

The FISTA iterations to find the sparse coefficients ($\overrightarrow{\mathcal{Z}}_t$) are

$$\overrightarrow{\mathcal{W}}_{t_{(j)}} = \overrightarrow{\mathcal{Y}}_{t_{(j-1)}} - \frac{1}{L(f)}\nabla_{\overrightarrow{\mathcal{Y}}}\left(\frac{1}{2}\|\mathcal{D}_t * \overrightarrow{\mathcal{Y}}_{t_{(j-1)}} - \overrightarrow{\mathcal{X}}_t\|_2^2\right) \tag{16}$$

$$\overrightarrow{\mathcal{Z}}_{t_{(j)}} = \mathcal{S}_{\lambda/L(f)}\left(\overrightarrow{\mathcal{W}}_{t_{(j)}}\right) \tag{17}$$

$$\gamma_{(j)} = \frac{1}{2}\left(1 + \sqrt{1 + 4\gamma_{(j-1)}^2}\right) \tag{18}$$

$$\overrightarrow{\mathcal{Y}}_{t_{(j)}} = \overrightarrow{\mathcal{Z}}_{t_{(j)}} + \frac{\gamma_{(j-1)} - 1}{\gamma_{(j)}}\left(\overrightarrow{\mathcal{Z}}_{t_{(j)}} - \overrightarrow{\mathcal{Z}}_{t_{(j-1)}}\right) \tag{19}$$

$\frac{1}{L(f)}$ acts as the step size of the gradient descent; $\gamma_0 = 1$, and $\mathcal{S}_{\lambda/L(f)}$ is the soft-thresholding operator. The soft-thresholding is applied element-wise to the tensor $\mathcal{S}_{\lambda/L(f)}(\overrightarrow{\mathcal{W}}_t) = \text{sign}(\overrightarrow{\mathcal{W}}_t) \odot \max(0, |\overrightarrow{\mathcal{W}}_t| - \frac{\lambda}{L(f)})$, where $\odot$ indicates element-wise multiplication. The momentum term provides faster convergence and is shown to have a convergence rate of $O\left(k^{-2}\right)$ [60], [61], where k counts the number of iterations. Computing the gradient of $f(\overrightarrow{\mathcal{Y}}_t)$ of Eq. (16) in the spatial domain is computationally expensive. We compute the gradient in the Fourier domain where $\overrightarrow{\bar{\mathcal{W}}}$, $\overrightarrow{\bar{\mathcal{X}}}$, $\bar{\mathcal{D}}$, $\overrightarrow{\bar{\mathcal{Z}}}$, and $\overrightarrow{\bar{\mathcal{Y}}}$ denote the FFT of $\overrightarrow{\mathcal{W}}_t$, $\overrightarrow{\mathcal{X}}_t$, $\mathcal{D}_t$, $\overrightarrow{\mathcal{Z}}_t$, and $\overrightarrow{\mathcal{Y}}_t$, respectively.

$$\overrightarrow{\bar{\mathcal{W}}}_{t_{(j)}} = \overrightarrow{\bar{\mathcal{Y}}}_{t_{(j-1)}} - \frac{1}{L(f)}\nabla_{\overrightarrow{\bar{\mathcal{Y}}}}\left(\frac{1}{2}\|\bar{\mathcal{D}}_t\overrightarrow{\bar{\mathcal{Y}}}_{t_{(j-1)}} - \overrightarrow{\bar{\mathcal{X}}}_t\|_2^2\right) \tag{20}$$

Algorithm 3 Foreground Detection

```
Input: $\vec{\mathcal{X}}_t \in \mathbb{R}^{n_1\times 1\times n_3}$,   $\bar{\mathcal{D}}_t \in \mathbb{C}^{n_1\times n_1\times n_3}$
       $\gamma^0 = 1$,   $\lambda > 0$
Output: $\vec{\mathcal{E}}_t$
 1: $\vec{\mathcal{X}}_t = \texttt{fft}(\vec{\mathcal{X}}_t, [\,], 3) \in \mathbb{C}^{n_1\times 1\times n_3}$
 2: while $\|\vec{\mathcal{Z}}_{t(j)} - \vec{\mathcal{Z}}_{t(j-1)}\|_2 > \epsilon$ do
 3:   for $i = 1 : \lceil\frac{n_3+1}{2}\rceil$ do
 4:     $\bar{\boldsymbol{w}}^{(i)}_{t(j)} = \bar{\boldsymbol{y}}^{(i)}_{t(j-1)} - \frac{1}{2}\bar{D}^{(i)}_t\left(\bar{\boldsymbol{y}}^{(i)}_{t(j-1)} - \bar{\boldsymbol{x}}^{(i)}_t\right)$
 5:   end for
 6:   for $i = \lceil\frac{n_3+1}{2}\rceil + 1 : n_3$ do
 7:     $\bar{\boldsymbol{w}}^{(i)}_{t(j)} = \texttt{conj}\left(\bar{\boldsymbol{w}}^{(n_3-i+2)}_{t(j)}\right)$
 8:   end for
 9:   $\vec{\mathcal{W}}_{t(j)} = \texttt{ifft}(\vec{\mathcal{W}}_{t(j)}, [\,], 3)$
10:   $\vec{\mathcal{Z}}_{t(j)} = \mathcal{S}_{\lambda/2}\left(\vec{\mathcal{W}}_{t(j)}\right)$
11:   $\vec{\mathcal{Z}}_{t(j)} = \texttt{fft}(\vec{\mathcal{Z}}_{t(j)}, [\,], 3)$
12:   $\gamma_{(j)} = \frac{1}{2}\left(1 + \sqrt{1 + 4\gamma^2_{(j-1)}}\right)$
13:   for $i = 1 : \lceil\frac{n_3+1}{2}\rceil$ do
14:     $\bar{\boldsymbol{y}}^{(i)}_{t(j)} = \bar{\boldsymbol{z}}^{(i)}_{t(j)} + \frac{\gamma_{(j-1)}-1}{\gamma_{(j)}}\left(\bar{\boldsymbol{z}}^{(i)}_{t(j)} - \bar{\boldsymbol{z}}^{(i)}_{t(j-1)}\right)$
15:   end for
16:   for $i = \lceil\frac{n_3+1}{2}\rceil + 1 : n_3$ do
17:     $\bar{\boldsymbol{y}}^{(i)}_{t(j)} = \texttt{conj}\left(\bar{\boldsymbol{y}}^{(n_3-i+2)}_{t(j)}\right)$
18:   end for
19:   $j = j + 1$
20: end while
21: for $i = 1 : \lceil\frac{n_3+1}{2}\rceil$ do
22:   $\bar{\boldsymbol{e}}^{(i)}_t = \bar{D}^{(i)}_t \bar{\boldsymbol{z}}^{(i)}_t$
23: end for
24: for $i = \lceil\frac{n_3+1}{2}\rceil + 1 : n_3$ do
25:   $\bar{\boldsymbol{e}}^{(i)}_t = \texttt{conj}\left(\bar{\boldsymbol{e}}^{(n_3-i+2)}_t\right)$
26: end for
27: $\vec{\mathcal{E}}_t = \texttt{ifft}(\vec{\mathcal{E}}_t, [\,], 3) \in \mathbb{R}^{n_1\times 1\times n_3}$
28: return $\vec{\mathcal{E}}_t$
```

The gradient of $f(\vec{\mathcal{Y}}_t)$ is computed as

$$\nabla\left(\frac{1}{2}\|\bar{\mathcal{D}}_t\vec{\mathcal{Y}}_t - \vec{\mathcal{X}}_t\|^2_2\right) = \bar{\mathcal{D}}^H_t\left(\bar{\mathcal{D}}_t\vec{\mathcal{Y}}_t - \vec{\mathcal{X}}_t\right) \quad (21)$$

We can further simplify the gradient function of Eq. (21) due to the unique structure of our dictionary $\bar{D}^{(i)H}_t\bar{D}^{(i)}_t = \bar{D}^{(i)}_t$ since $\left(I - \bar{U}^{(i)}_t\bar{U}^{(i)H}_t\right)^H\left(I - \bar{U}^{(i)}_t\bar{U}^{(i)H}_t\right) = I - \bar{U}^{(i)}_t\bar{U}^{(i)H}_t$ where $\bar{U}^{(i)}_t \in \mathbb{C}^{n_1\times d}$ consists of the truncated singular vectors of the frontal slice i in the frequency domain ($\bar{U}^{(i)H}_t\bar{U}^{(i)}_t = I$ and $\bar{U}^{(i)}_t\bar{U}^{(i)H}_t \neq I$). This can significantly speed-up the computation. The approximation of the Lipschitz constant $L(f)^{(i)}$ influences the performance of the FISTA. The Lipschitz constant is equal to the largest eigenvalue of $\bar{D}^{(i)H}_t\bar{D}^{(i)}_t$. This value is not easy to compute when the problem size is large. However, due to the unique structure of our

Algorithm 4 Tensor Subspace Tracking

```
Input: $\vec{\mathcal{L}}_t \in \mathbb{R}^{n_1\times 1\times n_3}$
Output: $\bar{\mathcal{U}}_t \in \mathbb{C}^{n_1\times d\times n_3}$
 1: $\vec{\mathcal{L}}_t = \texttt{fft}(\vec{\mathcal{L}}_t, [\,], 3)$
 2: for $i = 1 : \lceil\frac{n_3+1}{2}\rceil$ do
 3:   $\bar{\boldsymbol{j}}^{(i)}_t = \left(\bar{U}^{(i)}_{t-1}\right)^H \bar{\boldsymbol{l}}^{(i)}_t$
 4:   $\bar{\boldsymbol{k}}^{(i)}_t = \left(I - \bar{U}^{(i)}_{t-1}\bar{U}^{(i)H}_{t-1}\right)\bar{\boldsymbol{l}}^{(i)}_t$
 5:   $[\bar{Q}^{(i)}_t\ \bar{R}^{(i)}_t] = \texttt{qr}(\bar{\boldsymbol{k}}^{(i)}_t)$
 6:   $[\tilde{\bar{U}}^{(i)}_t\ \tilde{\bar{S}}^{(i)}_t\ \tilde{\bar{V}}^{(i)}_t] = \texttt{svd}\left(\begin{bmatrix}\bar{S}^{(i)}_{t-1} & \bar{\boldsymbol{j}}^{(i)}_t\\ 0 & \bar{R}^{(i)}_t\end{bmatrix}\right)$
 7:   $\bar{U}^{(i)}_t = \left[\bar{U}^{(i)}_{t-1}\ \bar{Q}^{(i)}_t\right]\tilde{\bar{U}}^{(i)}_t$
 8:   $\bar{S}^{(i)}_t = \tilde{\bar{S}}^{(i)}_t$
 9:   $\bar{V}^{(i)}_t = \begin{bmatrix}\bar{V}^{(i)}_{t-1} & 0\\ 0 & I\end{bmatrix}\tilde{\bar{V}}^{(i)}_t$
10: end for
11: for $i = \lceil\frac{n_3+1}{2}\rceil + 1 : n_3$ do
12:   $\bar{U}^{(i)}_t = \texttt{conj}\left(\bar{U}^{(n_3-i+2)}_t\right)$
13:   $\bar{S}^{(i)}_t = \bar{S}^{(n_3-i+2)}_t$
14:   $\bar{V}^{(i)}_t = \texttt{conj}\left(\bar{V}^{(n_3-i+2)}_t\right)$
15: end for
16: $\bar{\mathcal{U}}_t = \bar{\mathcal{U}}_t(:, 1:d, :)$
17: $\bar{\mathcal{S}}_t = \bar{\mathcal{S}}_t(1:d, 1:d, :)$
18: $\bar{\mathcal{V}}_t = \bar{\mathcal{V}}_t(:, 1:d, :)$
```

dictionary, we can obtain this value with no computational costs $L(f)^{(i)} = 2\|\bar{D}^{(i)}_t\|^2_2 = 2$ since $\|\bar{D}^{(i)}_t\|^2_2 = \|\bar{D}^{(i)}_t\|_2$.

$$\frac{1}{L(f)}\nabla f\left(\vec{\mathcal{Y}}_t\right) = \frac{1}{2}\bar{\mathcal{D}}_t\left(\vec{\mathcal{Y}}_t - \vec{\mathcal{X}}_t\right) \quad (22)$$

Equipped with this efficient gradient computation, we can write the FISTA iterations for our tensor convolutional sparse coding to detect the foreground as

$$\vec{\mathcal{W}}_{t(j)} = \vec{\mathcal{Y}}_{t(j-1)} - \frac{1}{2}\bar{\mathcal{D}}_t\left(\vec{\mathcal{Y}}_{t(j-1)} - \vec{\mathcal{X}}_t\right) \quad (23)$$

$$\vec{\mathcal{Z}}_{t(j)} = \mathcal{S}_{\lambda/2}\left(\vec{\mathcal{W}}_{t(j)}\right) \quad (24)$$

$$\gamma_{(j)} = \frac{1}{2}\left(1 + \sqrt{1 + 4\gamma^2_{(j-1)}}\right) \quad (25)$$

$$\vec{\mathcal{Y}}_{t(j)} = \vec{\mathcal{Z}}_{t(j)} + \frac{\gamma_{(j-1)} - 1}{\gamma_{(j)}}\left(\vec{\mathcal{Z}}_{t(j)} - \vec{\mathcal{Z}}_{t(j-1)}\right) \quad (26)$$

The stopping criteria is defined as a maximum number of iterations, or $\|\vec{\mathcal{Z}}_{t(j)} - \vec{\mathcal{Z}}_{t(j-1)}\|_2 \leq \epsilon$ where $\vec{\mathcal{Z}}_{t(j)}$ and $\vec{\mathcal{Z}}_{t(j-1)}$ are the last two iterations values of $\vec{\mathcal{Z}}$. A summary of our foreground detection method is given in Algorithm 3. Once the sparse coefficients ($\vec{\mathcal{Z}}_t$) are estimated, the low-rank background can be recovered as:

$$\vec{\mathcal{L}}_t = \vec{\mathcal{X}}_t - \mathcal{D}_t * \vec{\mathcal{Z}}_t \quad (27)$$

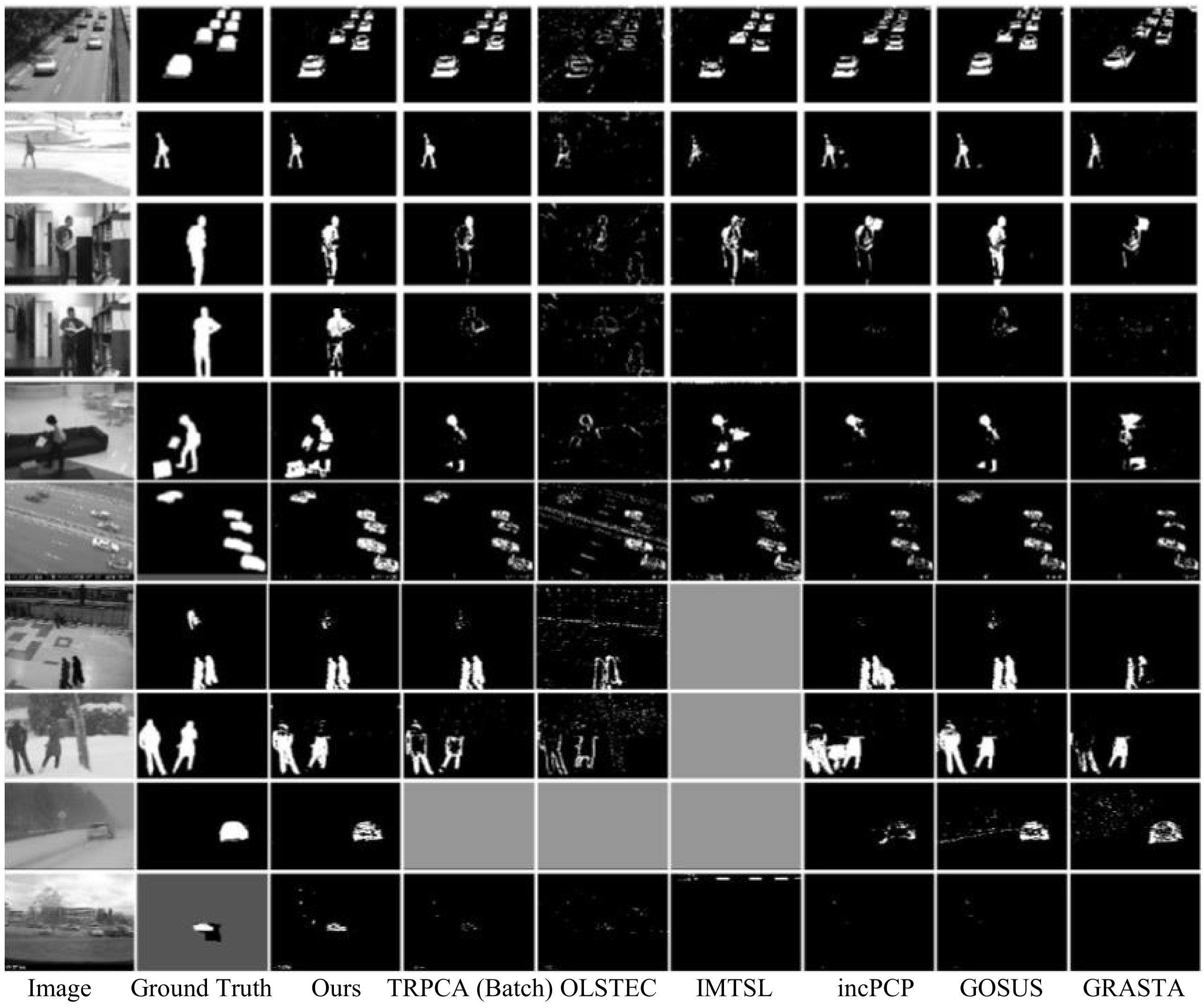

FIGURE 1. **Comparison of the detected foreground mask obtained by our algorithm and Batch Tensor RPCA [26], OLSTEC [53], IMTSL [55], incPCP [45], GOSUS [19] and GRASTA [43]. From the top 'Highway', 'Pedestrian', 'Office', 'Sofa', 'Turnpike', 'PETS2006', 'Skating', 'Snowfall' and 'Parking', respectively. We were unable to run IMTLS on 'PETS2006, 'Skating, and 'Snow Fall datasets due to the prolonged processing time (several hours per frame). Table 2 shows separation rate of each online RPCA method. Batch TRPCA and OLSTEC exceeded the memory limitation of 32 GB for 'Snowfall' dataset.**

The computational complexity of performing the Fourier transform on each incoming video frame is $O(n_1 n_3 \log n_3)$. To compute the gradient, we only perform matrix-to-vector products and the soft thresholding which take $O\left(n_3 n_1^2\right)$ and $O(n_1 n_3)$ flops, respectively. To update the basis tensor $\mathcal{U}_t \in \mathbb{R}^{n_1 \times d \times n_3}$, and subsequently, the dictionary $\mathcal{D}_t \in \mathbb{R}^{n_1 \times n_1 \times n_3}$. We extend the incremental SVD algorithm [63] to third-order tensors. The update is applied for every new incoming video frame to avoid performing a full T-SVD. The basic idea is to replace the full T-SVD with a series of much smaller SVD factorizations for the new incoming lateral slices $\overrightarrow{\mathcal{L}}_t \in \mathbb{R}^{n_1 \times 1 \times n_3}$. First, we compute the Fourier transform of the new incoming low-rank frame $\overrightarrow{\mathcal{L}}_t \in \mathbb{C}^{n_1 \times 1 \times n_3}$. We compute the projection of $\overrightarrow{\mathcal{L}}_t$ onto the orthogonal basis tensor $\bar{\mathcal{U}}_{t-1}$ and to the subspace orthogonal to $\bar{\mathcal{U}}_{t-1}$.

$$\bar{\boldsymbol{j}}_t^{(i)} = \bar{U}_{t-1}^{(i)\,H} \bar{\boldsymbol{l}}_t^{(i)} \tag{28}$$

$$\bar{\boldsymbol{k}}_t^{(i)} = \left(I - \bar{U}_{t-1}^{(i)} \bar{U}_{t-1}^{(i)\,H}\right) \bar{\boldsymbol{l}}_t^{(i)} \tag{29}$$

We perform QR factorization on $\bar{\boldsymbol{k}}_t^{(i)} = \bar{Q}_t^{(i)} \bar{R}_t^{(i)}$. We write each frontal slice of tensor $[\mathcal{L}_{t-1}, \ \overrightarrow{\mathcal{L}}_t]$ in the Fourier domain as:

$$\left[\bar{L}_{t-1}^{(i)} \quad \bar{\boldsymbol{l}}_t^{(i)}\right] = \left[\bar{U}_{t-1}^{(i)} \quad \bar{Q}_t^{(i)}\right] \begin{bmatrix} \bar{S}_{t-1}^{(i)} & \bar{\boldsymbol{j}}_t^{(i)} \\ 0 & \bar{R}_t^{(i)} \end{bmatrix} \begin{bmatrix} \bar{V}_{t-1}^{(i)} & 0 \\ 0 & I \end{bmatrix}^H \tag{30}$$

By taking the SVD of the middle matrix of (30), we have $\begin{bmatrix} \bar{S}_{t-1}^{(i)} & \bar{\boldsymbol{j}}_t^{(i)} \\ 0 & \bar{R}_t^{(i)} \end{bmatrix} = \tilde{\bar{U}}_t^{(i)} \tilde{\bar{S}}_t^{(i)} \tilde{\bar{V}}_t^{(i)H}$. The updated orthogonal tensor in the Fourier domain can be written as $\bar{U}_t^{(i)} = \left[\bar{U}_{t-1}^{(i)} \ \bar{Q}_t^{(i)}\right] \tilde{\bar{U}}_t^{(i)} \in \mathbb{C}^{n_1 \times (d+1) \times n_3}$. We truncate the estimated basis tensor by keeping the first d columns of $\bar{U}_t^{(i)}$. A summary of our tensor subspace tracking method is given

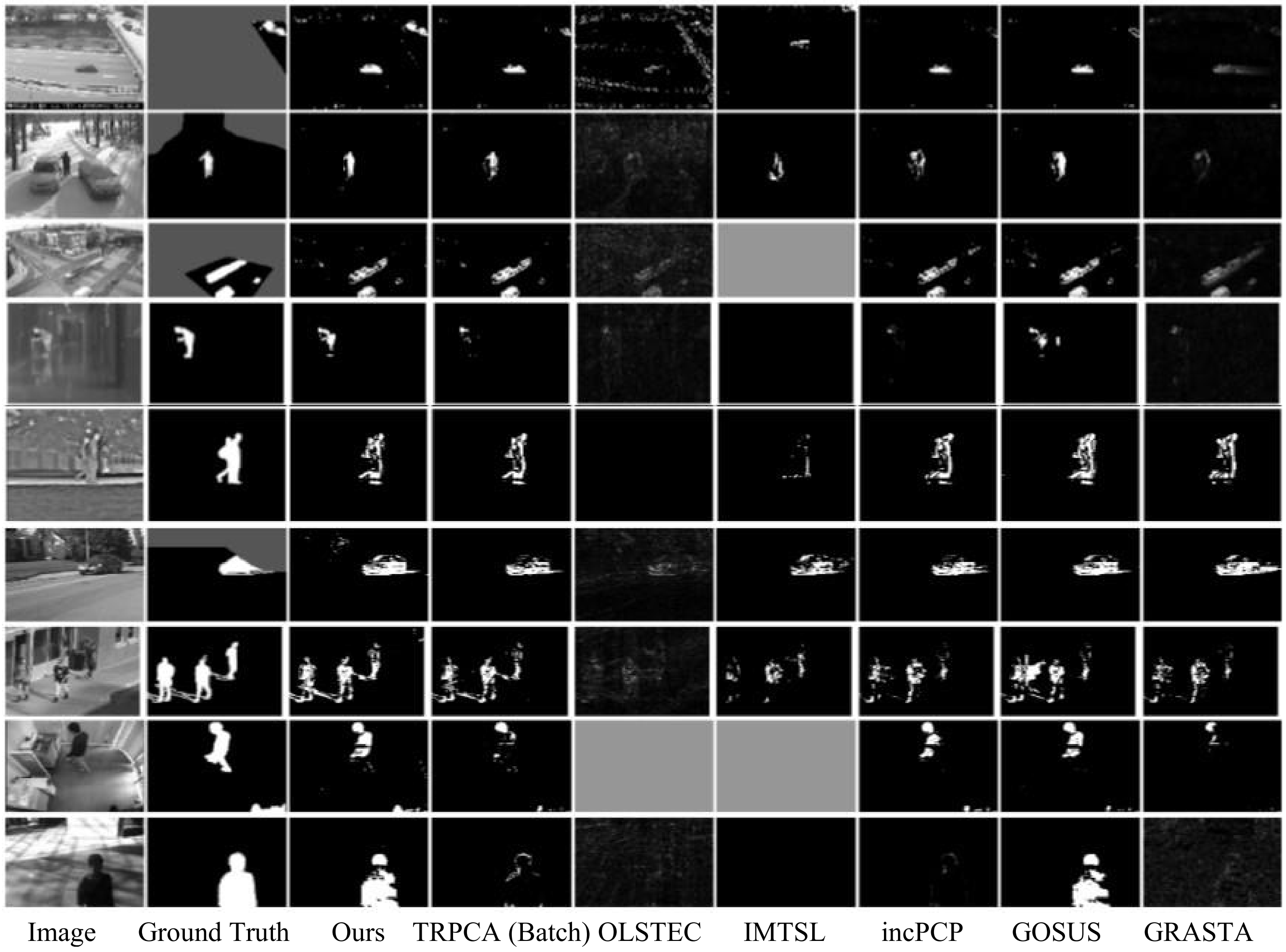

FIGURE 2. Comparison of the detected foreground mask obtained by our algorithm and Batch Tensor RPCA [26], OLSTEC [53], IMTSL [55], incPCP [45], GOSUS [19] and GRASTA [43]. From the top 'Streetlight', 'Winter driveway', 'Tram crossroad', 'Corridor', 'Park', 'Bungalow', 'Bus station', 'Copy machine', 'People shade', respectively. IMTLS did not complete foreground detection for 'Tram crossroad', and 'Copy machine' datasets due to extensive processing time (several hours per frame). Table 2 shows separation rate of each online RPCA method. OLSTEC exceeded the memory limitation of 32 GB for 'Copy machine' dataset.

in Algorithm 4. The tensor subspace tracking step has time complexity $O\left(d^2 n_3 n_1\right)$ where tubal rank-d is smaller than the size of the video frames.

V. RESULTS

We evaluated the performance of our algorithm[2] using 21 video sequences in the CDnet dataset [59] namely 'Highway', 'Office', 'Pedestrian', 'Sofa', 'Turnpike', 'PETS2006', 'Skating', 'Snowfall', 'Abandoned box', 'Parking', 'Streetlight', 'Winter driveway', 'Tram crossroad', 'Corridor', 'Dining room', 'Library', 'Park', 'Bungalow', 'Bus station', 'Copy machine', 'People shade'. The resolution of the video frames varies from 320×240 to 576×720. We also compared our algorithm's performance with existing online tensor RPCA algorithms namely IMTSL [55] and OLSTEC [53] and batch tensor RPCA [26] and three online matrix-based RPCA algorithms: GRASTA [43], incPCP [45], and GOSUS [19]. Fig. 1 shows ten snapshots of 'Highway', 'Pedestrian', 'Office', 'Sofa', 'Turnpike', 'PETS2006', 'Skating', 'Snowfall' and 'Parking' datasets. In 'Highway', 'Pedestrian' and 'PETS2006' datasets, the backgrounds remain approximately static over time; whereas, the foregrounds contain the moving parts (vehicles/people). The foreground masks are obtained by thresholding the foregrounds. The images, ground truth and foreground masks for each method are also displayed in Fig. 1. One notable example is the 'Office' video sequence in which a person enters a room at $t = 591$ and starts reading a book while standing in front of a camera. Our online tensor RPCA and GOSUS performed background layering; however, for other methods, the foregrounds were absorbed into the backgrounds at $t \in [640, 1900]$. In the 'Sofa' dataset, there are objects with intermittent motion. Our online tensor RPCA and IMTSL effectively detected the foreground as indicated in Fig. 1. However, other methods failed to detect the foreground when the part of the foreground remained static for a period of time and was absorbed into the background. For 'Turnpike' dataset, all methods could detect the foregrounds. However, OLSTEC suffers from a very high false positive rate.

[2]Codes of our method available at https://github.com/ESP-labGT

TABLE 1. F-score comparisons of RPCA methods for foreground detection. The best and second best performing online methods are shown in boldface and underlined, respectively.

	Ours	OLSTEC [53]	IMTSL [55]	incPCP [45]	GOSUS [19]	GRASTA [43]	Tensor RPCA (Batch) [26]
Highway	**0.82**	0.30	0.56	0.71	0.79	0.69	0.85
Office	**0.78**	0.19	0.28	0.43	0.57	0.32	0.46
Pedestrian	**0.88**	0.30	0.32	0.84	0.80	0.56	0.83
Sofa	**0.74**	0.18	0.28	0.40	0.40	0.27	0.55
Turnpike	**0.81**	0.46	0.64	0.77	**0.81**	**0.81**	0.83
PETS2006	**0.83**	0.14	†[a]	0.58	0.71	0.46	0.82
Skating	**0.81**	0.15	†[a]	0.62	0.79	0.58	0.56
Snowfall	**0.65**	‡[b]	†[a]	0.51	0.53	0.16	‡[b]
Abandoned box	**0.54**	0.10	†[a]	0.32	0.36	0.29	0.43
Parking	**0.69**	0.36	0.01	0.46	0.49	0.39	0.71
Streetlight	**0.89**	0.22	0.01	0.24	0.28	0.25	0.39
Winter driveway	**0.53**	0.02	0.13	0.40	0.37	0.12	0.52
Tram crossroad	**0.79**	0.42	†[a]	**0.79**	**0.79**	0.75	0.79
Corridor	**0.69**	0.21	0.03	0.56	0.55	0.24	0.41
Dining room	**0.77**	0.31	0.25	0.60	0.55	0.39	0.47
Library	**0.71**	0.39	0.04	0.34	0.39	0.35	0.39
Park	**0.69**	0.26	0.07	0.64	0.47	0.37	0.63
Bungalow	**0.66**	0.27	0.60	0.61	0.51	0.49	0.63
Bus station	**0.72**	0.23	0.29	0.57	0.55	0.42	0.63
Copy machine	**0.81**	‡[b]	†[a]	0.44	0.58	0.37	0.50
People shade	**0.86**	0.21	0.68	0.61	0.67	0.53	0.53
Average	**0.75**	–	–	0.54	0.57	0.42	–

[a]IMTSL did not complete foreground detection due to extensive processing time (several hours per frame).
[b]Algorithm operating on these datasets exceeded the memory limitation of 32 GB.

TABLE 2. Speed comparisons (seconds/frame) of online RPCA algorithms.

Dataset	Image size	Sequence length	Ours	OLSTEC [53]	IMTSL [55]	incPCP [45]	GOSUS [19]	GRASTA [43]
Highway	240×320	1700	0.1	0.82	745	0.13	4.9	0.01
Office	240×360	2050	0.11	1.1	960	0.14	4.1	0.01
Pedestrian	240×360	1099	0.11	0.6	973	0.14	4	0.01
Sofa	240×320	2750	0.1	1.3	738	0.14	3.4	0.01
Turnpike	240×320	1500	0.1	0.7	673	0.14	3.4	0.01
PETS2006	576×720	1200	2	7.6	31144	0.21	43.4	0.13
Skating	360×540	3900	0.6	7.7	5936	0.17	14.3	0.05
Snowfall	480×720	6500	1.6	‡[b]	21240	0.54	23.8	0.11
Abandoned box	288×432	4500	0.37	7	2078	0.09	7.4	0.03
Parking	240×320	2500	0.11	2.2	704	0.14	3.4	0.02
Streetlight	240×320	3200	0.12	3	694	0.11	5.6	0.03
Winter driveway	240×320	2500	0.15	1.3	682	0.17	4.4	0.02
Tram crossroad	350×640	900	0.7	0.7	7903	0.17	17.1	0.1
Corridor	240×320	5400	0.13	3.2	829	0.13	2.7	0.01
Dining room	240×320	3700	0.12	2.2	846	0.16	4.1	0.01
Library	240×320	4900	0.13	3.0	910	0.13	2.9	0.01
Park	288×352	600	0.26	0.5	1641	0.13	3.6	0.02
Bungalow	240×360	1700	0.13	1.2	1223	0.13	4.1	0.01
Bus station	240×360	1250	0.13	0.9	1128	0.13	4.2	0.01
Copy machine	480×720	3400	1.8	‡[b]	2365	0.2	22.7	0.09
People shade	244×380	1199	0.13	0.93	1344	1.7	5.1	0.02

Fig. 2 shows 9 snapshots from the 'Streetlight', 'Winter driveway', 'Tram crossroad', 'Corridor', 'Park', 'Bungalow', 'Bus station', 'Copy machine', 'People shade' video sequences. 'Corridor' and 'Park' provide examples from the thermal category. In the 'Corridor' dataset at $t = 1135$, a person is standing in front of a water fountain while drinking water. Only our method could effectively detect the foreground.

To quantitatively evaluate the performance of our online algorithm, we use F-measure as the accuracy metric [21], [31]. F-measure is defined as the harmonic mean of precision and recall: $F = \frac{2 \times Recall \times Precision}{Recall + Precision}$. *Recall* is the ratio of correctly classified foreground pixels (true positives (TP)) to the total number of foreground pixels in the ground truth $Recall = \frac{TP}{TP+FN}$. Precision gives the ratio of TP to the total pixels classified as foreground $Precision = \frac{TP}{TP+FP}$.

Table 1 shows the quantitative results in terms of F-measure of our algorithm as compared to the other online tensor-based and matrix-based RPCA algorithms. Our algorithm outperforms all online robust PCA in all categories. Fundamentally, batch methods should perform better since all data is available to them; whereas online methods receive data one frame per time instance. However, comparison of our method to batch tensor RPCA indicates that our method outperforms batch TRPCA in sixteen datasets, and the batch tensor RPCA performed better among 3 datasets. We were unable to run batch tensor RPCA for Snow Fall dataset ($480 \times 720 \times 6500$) since it exceeded our memory limit (32 GB).

Table 2 shows time per frame for online RPCA methods. All experiments are run on MATLAB 2021b using a computer with AMD Ryzen 7 2700X 3.70 GHz. Our algorithm achieves a separation rate of 0.1 second per frame for frame sizes 240×320. Our method performed background subtraction faster than the other online tensor RPCA. The processing speed of IMTLS drastically increases as frame size increases. IMTSL achieved a separation rate of $\sim$ 700 seconds per frame for frame sizes 240×320; and this processing time drastically increases to over 30000 second per frame for frame size 576×720. We were unable run IMTLS on PETS2006, Skating, Snow Fall, AbandonedBox, Tram crossroad, Copy machine datasets due to extraordinary processing times. The time complexity associated with the IMTSL method can be attributed to the low-rank decomposition. We should also note that OLSTEC is based on CP decomposition and IMTSL is based on Tucker decomposition while, our method is based on the T-SVD factorization. GRASTA performed faster than our online method. Our method performs background subtraction faster than GOSUS.

VI. CONCLUSION

In this paper, we proposed a new online tensor robust PCA algorithm for background subtraction based on the recently proposed T-product and T-SVD factorization. Our algorithm can be used to decompose a video frame into a low-rank background and a sparse foreground while preserving the multi-dimensional structures of data. We proposed a fast inverse-free sparse coding frame work to detect the foreground using fast iterative shrinkage-thresholding algorithm (FISTA). We developed an incremental tensor SVD algorithm to track the subspace changes. We assessed the performance of our algorithm on several real-world video datasets. Results indicate that our algorithm effectively detects the foreground, outperforming several existing state-of-the-art online RPCA algorithms.

REFERENCES

[1] G. Monteiro, J. Marcos, M. Ribeiro, and J. Batista, "Robust segmentation for outdoor traffic surveillance," in *Proc. 15th IEEE Int. Conf. Image Process.*, Oct. 2008, pp. 2652–2655.

[2] H. Huang, X. Fang, Y. Ye, S. Zhang, and P. L. Rosin, "Practical automatic background substitution for live video," *Comput. Vis. Media*, vol. 3, no. 3, pp. 273–284, 2017.

[3] S. Paul and A. K. Pal, "An efficient surveillance video coding scheme for static camera based captured video data," in *Proc. 6th Int. Conf. Signal Process. Integr. Netw. (SPIN)*, Mar. 2019, pp. 868–873.

[4] M. Alonso, A. Brunete, M. Hernando, and E. Gambao, "Background-subtraction algorithm optimization for home camera-based night-vision fall detectors," *IEEE Access*, vol. 7, pp. 152399–152411, 2019.

[5] E. J. Candès, X. Li, Y. Ma, and J. Wright, "Robust principal component analysis?" *J. ACM*, vol. 58, no. 1, pp. 1–37, 2009.

[6] L. Chen, X. Jiang, X. Liu, and Z. Zhou, "Robust low-rank tensor recovery via nonconvex singular value minimization," *IEEE Trans. Image Process.*, vol. 29, pp. 9044–9059, 2020.

[7] Y. Liu, L. Chen, and C. Zhu, "Improved robust tensor principal component analysis via low-rank core matrix," *IEEE J. Sel. Topics Signal Process.*, vol. 12, no. 6, pp. 1378–1389, Dec. 2018.

[8] J. Lou and Y. Cheung, "Robust low-rank tensor minimization via a new tensor spectral k-support norm," *IEEE Trans. Image Process.*, vol. 29, pp. 2314–2327, 2020.

[9] B. Qin, M. Jin, D. Hao, Y. Lv, Q. Liu, Y. Zhu, S. Ding, J. Zhao, and B. Fei, "Accurate vessel extraction via tensor completion of background layer in X-ray coronary angiograms," *Pattern Recognit.*, vol. 87, pp. 38–54, Mar. 2019.

[10] S. Ma and N. S. Aybat, "Efficient optimization algorithms for robust principal component analysis and its variants," *Proc. IEEE*, vol. 106, no. 8, pp. 1411–1426, Aug. 2018.

[11] R. Lin, X. Cao, Y. Xu, C. Wu, and H. Qiao, "Airborne moving vehicle detection for video surveillance of urban traffic," in *Proc. IEEE Intell. Vehicles Symp.*, Jun. 2009, pp. 203–208.

[12] H. Mansour and X. Jiang, "A robust online subspace estimation and tracking algorithm," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Apr. 2015, pp. 4065–4069.

[13] P. Narayanamurthy, V. Daneshpajooh, and N. Vaswani, "Provable subspace tracking from missing data and matrix completion," *IEEE Trans. Signal Process.*, vol. 67, no. 16, pp. 4245–4260, Aug. 2019.

[14] P. Narayanamurthy and N. Vaswani, "Provable dynamic robust PCA or robust subspace tracking," *IEEE Trans. Inf. Theory*, vol. 65, no. 3, pp. 1547–1577, Mar. 2019.

[15] H. Van Luong, N. Deligiannis, J. Seiler, S. Forchhammer, and A. Kaup, "Compressive online robust principal component analysis via n-ℓ_1 minimization," *IEEE Trans. Image Process.*, vol. 27, no. 9, pp. 4314–4329, Sep. 2018.

[16] H. Guo, C. Qiu, and N. Vaswani, "An online algorithm for separating sparse and low-dimensional signal sequences from their sum," *IEEE Trans. Signal Process.*, vol. 62, no. 16, pp. 4284–4297, Aug. 2014.

[17] C. Qiu, N. Vaswani, B. Lois, and L. Hogben, "Recursive robust PCA or recursive sparse recovery in large but structured noise," *IEEE Trans. Inf. Theory*, vol. 60, no. 8, pp. 5007–5039, Aug. 2014.

[18] J. Zhan and N. Vaswani, "Robust PCA with partial subspace knowledge," *IEEE Trans. Signal Process.*, vol. 63, no. 13, pp. 3332–3347, Jul. 2015.

[19] J. Xu, V. K. Ithapu, L. Mukherjee, J. M. Rehg, and V. Singh, "GOSUS: Grassmannian online subspace updates with structured-sparsity," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2013, pp. 3376–3383.

[20] N. Vaswani and P. Narayanamurthy, "Static and dynamic robust PCA and matrix completion: A review," *Proc. IEEE*, vol. 106, no. 8, pp. 1359–1379, Aug. 2018.

[21] N. Vaswani, T. Bouwmans, S. Javed, and P. Narayanamurthy, "Robust subspace learning: Robust PCA, robust subspace tracking, and robust subspace recovery," *IEEE Signal Process. Mag.*, vol. 35, no. 4, pp. 32–55, Jul. 2018.

[22] 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.

[23] A. Cichocki, D. Mandic, L. D. Lathauwer, G. Zhou, Q. Zhao, C. Caiafa, and H. A. Phan, "Tensor decompositions for signal processing applications: From two-way to multiway component analysis," *IEEE Signal Process. Mag.*, vol. 32, no. 2, pp. 145–163, Mar. 2015.

[24] A. H. Phan and A. Cichocki, "Tensor decompositions for feature extraction and classification of high dimensional datasets," *IEICE Nonlinear Theory Appl.*, vol. 1, no. 1, pp. 37–68, 2010.

[25] N. D. Sidiropoulos, L. De Lathauwer, X. Fu, K. Huang, E. E. Papalexakis, and C. Faloutsos, "Tensor decomposition for signal processing and machine learning," *IEEE Trans. Signal Process.*, vol. 65, no. 13, pp. 3551–3582, Jul. 2017.

[26] 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, Jan. 2020.

[27] B. Shen and Kong, "Robust tensor principal component analysis: Exact recovery via deterministic model," 2020, *arXiv:2008.02211*.

[28] 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.

[29] Y. Zhou and Y.-M. Cheung, "Bayesian low-tubal-rank robust tensor factorization with multi-rank determination," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 1, pp. 62–76, Jan. 2021.

[30] P. Zhou, C. Lu, Z. Lin, and C. Zhang, "Tensor factorization for low-rank tensor completion," *IEEE Trans. Image Process.*, vol. 27, no. 3, pp. 1152–1163, Mar. 2017.

[31] X. Liu, G. Zhao, J. Yao, and C. Qi, "Background subtraction based on low-rank and structured sparse decomposition," *IEEE Trans. Image Process.*, vol. 24, no. 8, pp. 2502–2514, Aug. 2015.

[32] S. E. Ebadi and E. Izquierdo, "Foreground segmentation with tree-structured sparse RPCA," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 9, pp. 2273–2280, Sep. 2017.

[33] S. Wang, Y. Wang, Y. Chen, P. Pan, Z. Sun, and G. He, "Robust PCA using matrix factorization for background/foreground separation," *IEEE Access*, vol. 6, pp. 18945–18953, 2018.

[34] S. Javed, A. Mahmood, T. Bouwmans, and S. K. Jung, "Spatiotemporal low-rank modeling for complex scene background initialization," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 28, no. 6, pp. 1315–1329, Jun. 2018.

[35] T. Bouwmans, S. Javed, H. Zhang, Z. Lin, and R. Otazo, "On the applications of robust PCA in image and video processing," *Proc. IEEE*, vol. 106, no. 8, pp. 1427–1457, Aug. 2018.

[36] T. Bouwmans and E. H. Zahzah, "Robust PCA via principal component pursuit: A review for a comparative evaluation in video surveillance," *Comput. Vis. Image Understand.*, vol. 122, pp. 22–34, May 2014.

[37] T. Bouwmans, F. Porikli, B. Höferlin, and A. Vacavant, *Background Modeling and Foreground Detection for Video Surveillance*. Boca Raton, FL, USA: CRC Press, 2014.

[38] T. Bouwmans, A. Sobral, S. Javed, S. K. Jung, and E.-H. Zahzah, "Decomposition into low-rank plus additive matrices for background/foreground separation: A review for a comparative evaluation with a large-scale dataset," *Comput. Sci. Rev.*, vol. 23, pp. 1–71, Feb. 2017.

[39] T. Bouwmans, C. Silva, C. Marghes, M. S. Zitouni, H. Bhaskar, and C. Frelicot, "On the role and the importance of features for background modeling and foreground detection," *Comput. Sci. Rev.*, vol. 28, pp. 26–91, May 2018.

[40] B. Garcia-Garcia, T. Bouwmans, and A. J. R. Silva, "Background subtraction in real applications: Challenges, current models and future directions," *Comput. Sci. Rev.*, vol. 35, Feb. 2020, Art. no. 100204.

[41] T. Bouwmans, N. S. Aybat, and E.-H. Zahzah, *Handbook of Robust Low-Rank and Sparse Matrix Decomposition: Applications in Image and Video Processing*. Boca Raton, FL, USA: CRC Press, 2016.

[42] S. Javed, A. Mahmood, T. Bouwmans, and S. K. Jung, "Background–foreground modeling based on spatiotemporal sparse subspace clustering," *IEEE Trans. Image Process.*, vol. 26, no. 12, pp. 5840–5854, Dec. 2017.

[43] J. He, L. Balzano, and A. Szlam, "Incremental gradient on the Grassmannian for online foreground and background separation in subsampled video," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2012, pp. 1568–1575.

[44] F. Seidel, C. Hage, and M. Kleinsteuber, "pROST: A smoothed ℓ_p-norm robust online subspace tracking method for background subtraction in video," *Mach. Vis. Appl.*, vol. 25, no. 5, pp. 1227–1240, 2014.

[45] P. Rodriguez and B. Wohlberg, "Incremental principal component pursuit for video background modeling," *J. Math. Imag. Vis.*, vol. 55, no. 1, pp. 1–18, 2016.

[46] S. Javed, S. H. Oh, A. Sobral, T. Bouwmans, and S. K. Jung, "Background subtraction via superpixel-based online matrix decomposition with structured foreground constraints," in *Proc. IEEE Int. Conf. Comput. Vis. Workshop (ICCVW)*, Dec. 2015, pp. 930–938.

[47] A. Anandkumar, P. Jain, Y. Shi, and U. N. Niranjan, "Tensor vs. matrix methods: Robust tensor decomposition under block sparse perturbations," 2016, *arXiv:1510.04747*.

[48] A. Ozdemir, E. M. Bernat, and S. Aviyente, "Recursive tensor subspace tracking for dynamic brain network analysis," *IEEE Trans. Signal Inf. Process. Over Netw.*, vol. 3, no. 4, pp. 669–682, Dec. 2017.

[49] L. Li, W. Huang, I. Y.-H. Gu, and Q. Tian, "Statistical modeling of complex backgrounds for foreground object detection," *IEEE Trans. Image Process.*, vol. 13, no. 11, pp. 1459–1472, Nov. 2004.

[50] P. Li, J. Feng, X. Jin, L. Zhang, X. Xu, and S. Yan, "Online robust low-rank tensor learning," in *Proc. 26th Int. Joint Conf. Artif. Intell.*, Aug. 2017, pp. 2180–2186.

[51] K. Mahapatra and N. R. Chaudhuri, "Online robust PCA for malicious attack-resilience in wide-area mode metering application," *IEEE Trans. Power Syst.*, vol. 34, no. 4, pp. 2598–2610, Jul. 2019.

[52] H. Kasai, "Fast online low-rank tensor subspace tracking by CP decomposition using recursive least squares from incomplete observations," *Neurocomputing*, vol. 347, pp. 177–190, Jun. 2019.

[53] H. Kasai, "Online low-rank tensor subspace tracking from incomplete data by CP decomposition using recursive least squares," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Mar. 2016, pp. 2519–2523.

[54] Z. Zhang, D. Liu, S. Aeron, and A. Vetro, "An online tensor robust PCA algorithm for sequential 2D data," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Mar. 2016, pp. 2434–2438.

[55] A. Sobral, C. G. Baker, T. Bouwmans, and E.-H. Zahzah, "Incremental and multi-feature tensor subspace learning applied for background modeling and subtraction," in *Proc. Int. Conf. Image Anal. Recognit.* Cham, Switzerland: Springer, 2014, pp. 94–103.

[56] 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, Feb. 2013.

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

[58] 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, Jul. 2021, Art. no. e2015851118.

[59] Y. Wang, P.-M. Jodoin, F. Porikli, J. Konrad, Y. Benezeth, and P. Ishwar, "CDnet 2014: An expanded change detection benchmark dataset," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops*, Jun. 2014, pp. 387–394.

[60] A. Beck and M. Teboulle, "A fast iterative shrinkage-thresholding algorithm for linear inverse problems," *SIAM J. Imag. Sci.*, vol. 2, no. 1, pp. 183–202, 2009.

[61] A. Beck and M. Teboulle, "A fast iterative shrinkage-thresholding algorithm with application to wavelet-based image deblurring," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, Apr. 2009, pp. 693–696.

[62] C. Garcia-Cardona and B. Wohlberg, "Convolutional dictionary learning: A comparative review and new algorithms," *IEEE Trans. Comput. Imag.*, vol. 4, no. 3, pp. 366–381, Sep. 2018.

[63] M. Brand, "Incremental singular value decomposition of uncertain data with missing values," in *Computer Vision—ECCV*, A. Heyden, G. Sparr, M. Nielsen, P. Johansen, Eds. Berlin, Germany: Springer, 2002, pp. 707–720.

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 Institute of Technology. His research interests include audio and psychoacoustics, signal processing in the context of human perception, and efficient and real-time application of such techniques using both analog and digital hardware. In 2004, he was awarded the National Science Foundation Career Award for excellence as a young educator and researcher and the Presidential Early Career Award for scientists and engineers.