Robust Tensor Decomposition based Background/Foreground Separation in Noisy Videos and Its Applications in Additive Manufacturing

Bo Shen (*Graduate Student Member*), Rakesh R. Kamath, Hahn Choo, Zhenyu (James) Kong (*Member*)

Abstract—Background/foreground separation is one of the most fundamental tasks in computer vision, especially for video data. Robust PCA (RPCA) and its tensor extension, namely, Robust Tensor PCA (RTPCA), provide an effective framework for background/foreground separation by decomposing the data into low-rank and sparse components, which contain the background and the foreground (moving objects), respectively. However, in real-world applications, the video data is contaminated with noise. For example, in metal additive manufacturing (AM), the processed X-ray video to study melt pool dynamics is very noisy. RPCA and RTPCA are not able to separate the background, foreground, and noise simultaneously. As a result, the noise will contaminate the background or the foreground or both. There is a need to remove the noise from the background and foreground. To achieve the three terms decomposition, a smooth sparse Robust Tensor Decomposition (SS-RTD) model is proposed to decompose the data into static background, smooth foreground, and noise, respectively. Specifically, the static background is modeled by the low-rank Tucker decomposition, the smooth foreground (moving objects) is modeled by the spatio-temporal continuity, which is enforced by the total variation regularization, and the noise is modeled by the sparsity, which is enforced by the ℓ_1 norm. An efficient algorithm based on alternating direction method of multipliers (ADMM) is implemented to solve the proposed model. Extensive experiments on both simulated and real data demonstrate that the proposed method significantly outperforms the state-of-the-art approaches for background/foreground separation in noisy cases.

Note to Practitioners—This work is motivated by melt pool detection in metal additive manufacturing where the processed X-ray video from the monitoring system is very noisy. The objective is to recover the background with porosity defects and the foreground with melt pool in the presence of noise. Existing methods fail to separate the noise from the background and foreground since RPCA and RTPCA have only two components, which cannot explain the three components in the data. This paper puts forward a smooth sparse Robust Tensor Decomposition by decomposing the tensor data into low-rank, smooth, and sparse components, respectively. It is a highly effective method for background/foreground separation in noisy case. In the case studies on simulated video and X-ray data, the proposed method can handle non-additive noise, and even the case of high noise-ratio. In the proposed algorithm, there is only one tuning parameter λ . Based on the case studies, our method achieves satisfying performance by taking any $\lambda \in [0.2, 1]$ with anisotropic total variation regularization ($q = \{1\}$). That means the proposed method is a kind of “parameter-free”

algorithm. Furthermore, the proposed method is also applicable to other popular industrial applications. Practitioners can use the proposed SS-RTD for degradation processes monitoring, where the degradation image contains the static background, anomaly, and random disturbance, respectively.

Index Terms—Robust Tensor Decomposition (RTD), Smooth Sparse Decomposition, Spatio-temporal Continuity, Total Variation Regularization, Low-rankness.

NOMENCLATURE

H, W, T	The height, width, and number of an image frame
(r_1, r_2, r_3)	The multi-linear rank in Tucker Decomposition
λ	The balance coefficient in the proposed objective function
$\mathcal{X}$	The order three tensor in $\mathbb{R}^{H \times W \times T}$ represented by $\{\mathbf{X}_1, \dots, \mathbf{X}_T\}$
$\mathbf{X}_t$	t -th image frame in $\mathbb{R}^{H \times W}$
$\mathcal{L}$	The low-rank tensor (static video background)
$\mathcal{S}$	The smooth tensor (smooth moving objects)
$\mathcal{E}$	The noise tensor (all kinds of noise)
$\mathcal{X} \times_n \mathbf{U}$	The mode- n multiplication of a tensor $\mathcal{X}$ with a matrix $\mathbf{U}$
$\mathcal{G}$	The core tensor in Tucker decomposition
$\mathbf{U}_1, \mathbf{U}_2, \mathbf{U}_3$	The factor matrices in Tucker decomposition
f	The auxiliary variable
$\mathbf{D}_h, \mathbf{D}_v, \mathbf{D}_t$	Three vectorizations of the difference operation along the horizontal, vertical, and temporal directions
$\mathbf{D}$	The concatenated difference operation, i.e., $[\mathbf{D}_h^\top, \mathbf{D}_v^\top, \mathbf{D}_t^\top]^\top$
$\ \cdot\ _F$	The Frobenius norm
$\ \cdot\ _1$	The ℓ_1 norm
$\ \cdot\ _2$	The ℓ_2 norm
$\ \cdot\ _{TV1}$	The anisotropic total variation norm
$\ \cdot\ _{TV2}$	The isotropic total variation norm
Q	$Q \in \{TV1, TV2\}$
q	$q = \{1\}$ corresponding to the anisotropic total variation norm; $q = \{2, 1\}$ corresponding to the isotropic total variation norm
$\text{vec}(\cdot)$	The vectorization operator
$\text{ten}(\cdot)$	The tensorization operator
$\boldsymbol{\lambda}^f, \boldsymbol{\Lambda}^{\mathcal{X}}$	The Lagrange multiplier vector and tensor
$\beta^f, \beta^{\mathcal{X}}$	The positive penalty scalars
c_1, c_2	The coefficients in adaptive updating scheme for $\beta^f, \beta^{\mathcal{X}}$

Corresponding author: Zhenyu (James) Kong.

B. Shen, Z.J. Kong are with the Grado Department of Industrial and Systems Engineering, Virginia Tech, Blacksburg, VA 24061 USA; e-mail: (boshen@vt.edu; zkong@vt.edu).

R.R. Kamath, H. Choo are with the Department of Materials Science and Engineering, The University of Tennessee, Knoxville, TN 37996 USA; e-mail: (rkamath@vols.utk.edu; hchoo@utk.edu).

Fig. 1. (a) Unprocessed X-ray image; (b) Processed X-ray image; (c) Processed X-ray image with manually annotated solid/liquid interface (Yellow rectangle boxes in (a)~(c) are the regions of the melt pool, the red boundary in (c) is the solid/liquid interface).

fftn	The fast 3D Fourier transform
ifftn	The inverse fast 3D Fourier transform
$\text{soft}(\cdot, \cdot)$	The soft-thresholding operator
γ	The parameter associated with convergence rate in ADMM
$\text{Err}(\cdot)$	The error of the auxiliary variable
relChgA	The relative change of $\mathcal{A}$
relErrA	The relative error of $\mathcal{A}$

I. INTRODUCTION

BACKGROUND/Foreground separation is a fundamental step for moving object detection in many video data applications [1], [2]. It is usually performed by separating the moving objects called “foreground” from the static objects called “background” [3], [4]. In many real-world applications, the presence of noise is a very common but challenging issue [5], [6]. For example, in metal additive manufacturing (AM) research, high-speed X-ray data has been applied to study the melt pool formation and evolution [7]. The change of melt pool geometry is directly associated with solid/liquid interface velocity, which can be utilized to characterize the microstructure of the final part [8]. To capture the melt pool, a high-speed X-ray imaging system is applied to monitor the printing process [9]. One unprocessed X-ray image from the monitoring system is shown in Fig. 1a, where the melt pool cannot be identified by our naked eyes. To enhance the boundary of the melt pool, the unprocessed X-ray image is first subtracted by the X-ray image captured six frames earlier and then adjusted contrast to obtain the processed X-ray image. The processed X-ray image is shown in Fig. 1b, where the melt pool boundary can be located by our naked eyes as outlined in red shown in Fig. 1c. Even though the melt pool boundary is enhanced, this subtraction step generates lots of random noise causing problems for accurate boundary detection.

Recent research on background/foreground separation is based on decomposition of video data into low-rank and sparse components. It is an effective framework to separate the foreground from the background, which are modeled by the sparse and low-rank components, respectively. Among them, the most representative problem formulation is the Robust Principal Component Analysis (RPCA) [3], [10], which is a modification of the widely used statistical procedure named principal component analysis (PCA). RPCA [10] decomposes the data matrix $\mathbf{X}$ into the sum of a low-rank matrix $\mathbf{L}$ and

a sparse matrix $\mathbf{S}$, where the low-rankness and sparsity are measured by the nuclear norm $\|\cdot\|_*$ and ℓ_1 norm $\|\cdot\|_1$, respectively. Apart from the background/foreground separation, RPCA can also be applied to the task of image denoising [2], where the noise is represented by the sparse components.

One major disadvantage of RPCA is that it can only deal with 2-D matrix data since the nuclear norm $\|\cdot\|_*$ is designed for matrix. However, real-world data is usually multi-dimensional in nature, where rich information is stored in multi-way arrays known as tensors [11]. For example, a greyscale video is 3-D data, which stacks multiple images along the time domain; a color image is also 3-D data that has three channels: red, green, and blue, where each channel is a 2-D image. To apply RPCA to these datasets, the multi-way tensor data has to be reconstructed into a matrix. Such a preprocessing usually leads to information loss and performance degradation since the structure information in the data is deteriorated. To address this issue, it is necessary to consider extending RPCA to manipulate the tensor data directly by taking advantage of its multi-dimensional structure. However, it is challenging to do so since the numerical algebra of tensors is fraught with many computationally hard problems [12], [13].

Contributed by the newly developed tensor multiplication scheme on t-SVD [14], Zhang *et al.* [15] proposed the tensor tubal rank as well as the tensor nuclear norm for image denoising. Based on the tensor nuclear norm, Lu *et al.* [16] developed Robust Tensor PCA (RTPCA) by extending RPCA from 2-D matrix to 3-D tensor data, aiming to exactly recover a low-rank tensor contaminated by sparse errors. More specifically, it tries to recover the low-rank tensor $\mathcal{L}$ and sparse tensor $\mathcal{E}$ from the data tensor $\mathcal{X}$, which can be represented $\mathcal{X} = \mathcal{L} + \mathcal{E}$. To further improve RTPCA, some work has been proposed with different objective functions and constraints [17]–[19]. However, none of them is able to address the background/foreground separation problem in noisy cases because the low-rank and sparse components extracted from RPCA or RTPCA algorithms do not consider the noise components in their analyses.

The objective of this study is to address the problem of background/foreground separation in the presence of noise and apply it to additive manufacturing applications (Figure 1). To achieve this objective, a smooth sparse Robust Tensor Decomposition (SS-RTD) is proposed to decompose the data

tensor $\mathcal{X}$ into a low-rank tensor (background) $\mathcal{L}$, a smooth tensor (foreground) $\mathcal{S}$, and a sparse tensor (noise) $\mathcal{E}$, namely, $\mathcal{X} = \mathcal{L} + \mathcal{S} + \mathcal{E}$. In the SS-RTD, the background is modeled by the low-rank Tucker decomposition [11]. The spatio-temporal continuity is applied to formulate the moving objects (foreground) [20], [21]. That is, the moving objects in video foreground are spatially continuous in both its support regions and its intensity values in these regions. Moreover, the moving objects are also temporally continuous among succeeding frames. The noise is modeled by a sparse tensor. To summarize, the contributions of this paper are as follows:

- (i) Propose the smooth sparse Robust Tensor Decomposition model for background/foreground separation in the presence of noise by decomposing the data tensor into a low-rank tensor, a smooth tensor, and a sparse tensor, respectively;
- (ii) Implement an efficient algorithm based on alternating direction method of multipliers (ADMM) [22] to solve the proposed model.

The remainder of this paper is organized as follows. A brief review of notation and related research work is provided in Section II. The proposed model and algorithm to solve this model are introduced in Section III. Numerical studies in Section IV and real-world additive manufacturing application in Section V are provided for testing and validation of the proposed method. Finally, the conclusions and future work are discussed in Section VI.

II. NOTATION AND RESEARCH BACKGROUND

In Section II-A, the notation and basics in multi-linear algebra used in this paper are reviewed. Then, the smooth sparse decomposition and Robust Tensor PCA are reviewed briefly in Section II-B. Afterwards, the research gaps of the existing work are identified in Section II-C.

A. Notation and Tensor Basis

Throughout this paper, scalars are denoted by lowercase letters, e.g., x ; vectors are denoted by lowercase boldface letters, e.g., $\mathbf{x}$; matrices are denoted by uppercase boldface, e.g., $\mathbf{X}$; and tensors are denoted by calligraphic letters, e.g., $\mathcal{X}$. The order of a tensor is the number of its modes or dimensions. A real-valued tensor of order N is denoted by $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \dots \times I_N}$ and its entries by $\mathcal{X}(i_1, i_2, \dots, i_N)$. The multi-linear Tucker rank of an N -order tensor is the tuple of the ranks of the mode- n unfoldings $\mathbf{X}_{(n)} \in \mathbb{R}^{I_n \times (I_1 \times \dots \times I_{n-1} \times I_{n+1} \times \dots \times I_N)}$. The inner product of two same-sized tensors $\mathcal{X}$ and $\mathcal{Y}$ is the sum of the products of their entries, i.e., $\langle \mathcal{X}, \mathcal{Y} \rangle = \sum_{i_1} \dots \sum_{i_N} \mathcal{X}(i_1, \dots, i_N) \cdot \mathcal{Y}(i_1, \dots, i_N)$. Following the definition of inner product, the Frobenius norm of a tensor $\mathcal{X}$ is defined as $\|\mathcal{X}\|_F = \sqrt{\langle \mathcal{X}, \mathcal{X} \rangle}$. The mode- n multiplication of a tensor $\mathcal{X}$ with a matrix $\mathbf{U}$ amounts to the multiplication of all mode- n vector fibers with $\mathbf{U}$, i.e., $(\mathcal{X} \times_n \mathbf{U})(i_1, \dots, i_{n-1}, j_n, i_{n+1}, \dots, i_N) = \sum_{i_n} \mathcal{X}(i_1, \dots, i_N) \cdot \mathbf{U}(j_n, i_n)$.

B. Related Work

In this subsection, two directions of related work to motivate the research in this paper are introduced here.

1) *Smooth Sparse Decomposition*: In the literature of anomaly detection in images, there is a popular two-steps procedure, namely, smoothing images first followed by anomaly detection [23]. However, this procedure may not be optimal. To address this issue, Yan *et al.* [24] proposed a smooth sparse decomposition (SSD) approach to integrate smoothing images and anomaly detection into a single task. It decomposes an image $\mathbf{y}$ into a smooth image background $\boldsymbol{\mu}$, sparse anomalous regions $\mathbf{a}$, and random noise $\mathbf{e}$, namely, $\mathbf{y} = \boldsymbol{\mu} + \mathbf{a} + \mathbf{e}$. To further enhance the model, the authors proposed to use smooth spline basis $\mathbf{B}$ and sparse spline basis $\mathbf{B}_a$ to formulate the mean and anomalies. Specifically, the enhanced SSD model is as $\mathbf{y} = \mathbf{B}\boldsymbol{\theta} + \mathbf{B}_a\boldsymbol{\theta}_a + \mathbf{e}$, where $\boldsymbol{\theta}$ is the smooth basis coefficients and $\boldsymbol{\theta}_a$ is the sparse basis coefficients, respectively. To estimate $\boldsymbol{\theta}$ and $\boldsymbol{\theta}_a$, a least square regression penalized by the two terms is as follows

$$\begin{aligned} \min_{\boldsymbol{\theta}, \boldsymbol{\theta}_a} \quad & \|\mathbf{e}\|_2^2 + \tau_1 \boldsymbol{\theta}^\top \mathbf{A} \boldsymbol{\theta} + \tau_2 \|\boldsymbol{\theta}_a\|_1 \\ \text{s.t.} \quad & \mathbf{y} = \mathbf{B}\boldsymbol{\theta} + \mathbf{B}_a\boldsymbol{\theta}_a + \mathbf{e}, \end{aligned}$$

where $\|\mathbf{e}\|_2^2$ is the fitting error; the first penalty term measures the smoothness in $\boldsymbol{\theta}$ through the roughness matrix $\mathbf{A}$; the second penalty term describes the sparsity in $\boldsymbol{\theta}_a$; and $\tau_1, \tau_2 > 0$ are regularization coefficients to be tuned. The idea of smooth and sparse decomposition is further extended to the literature of tensor completion [25].

2) *Robust Tensor PCA*: As the tensor extension of the popular Robust PCA, the recent proposed RTPCA [16] aims to recover the low rank tensor $\mathcal{L}_0 \in \mathbb{R}^{I_1 \times I_2 \times I_3}$ and sparse tensor $\mathcal{E}_0 \in \mathbb{R}^{I_1 \times I_2 \times I_3}$ from their sum. RTPCA solves the following convex optimization problem

$$\min_{\mathcal{L}, \mathcal{E}} \|\mathcal{L}\|_{TNN} + \rho \|\mathcal{E}\|_1, \text{ s.t. } \mathcal{X} = \mathcal{L} + \mathcal{E},$$

where $\|\cdot\|_{TNN}$ is their proposed tensor nuclear norm, which is a convex relaxation of the tensor tubal rank. The tensor nuclear norm and tensor tubal rank are defined based on the t-SVD proposed in [15]. In the paper, the analysis shows that $\rho = 1/\sqrt{\max(I_1, I_2)I_3}$ guarantees the exact recovery under tensor incoherence condition. Following this direction, to further exploit the low-rank structures in tensor data, Liu *et al.* [17] extracted low-rank component for the core matrix whose entries are from the diagonal elements of the core tensor. Based on this idea, they defined a new tensor nuclear norm and proposed a creative algorithm to deal with RTPCA problems. Other than the work based on the tensor tubal rank, Yang *et al.* [19] considered a new model for RTPCA based on tensor train rank. These methods are applied to background/foreground separation, image/video denosing, etc.

C. Research Gap Identification

In real-world applications, the video data is often contaminated with noise during acquisition, compression, and transmission [6] exemplified in Fig. 1. If the smooth sparse decomposition method [24] is applied to the noisy video,

Fig. 2. Illustration for the decomposition strategy of a video in the proposed method.

the background and foreground cannot be separated well since they together are considered as smooth components and the noise is considered as sparse components. If the Robust Tensor PCA [17]–[19] is applied, either the detected moving objects or the background is contaminated with noise since RTPCA cannot handle the three components simultaneously (i.e., background, foreground, and noise). Therefore, this work seeks to address these research gaps by devising a new smooth sparse Robust Tensor Decomposition (SS-RTD). The proposed model can be considered as separating background/foreground together with video denoising by providing a new decomposing methodology with three components.

III. PROPOSED METHOD

In Section III-A, the proposed smooth sparse RTD for background/foreground separation in the presence of noise is presented. Specifically, the low-rankness, spatio-temporal continuity, and sparsity are formulated by the Tucker decomposition, total variation (TV) regularization, and ℓ_1 regularization, respectively. In Section III-B, an efficient algorithm based on ADMM [22] is designed to solve the proposed model.

A. Proposed Smooth Sparse RTD Model

Throughout this work, it is focused on the video that can be represented as a third-order tensor $\mathcal{X} := \{\mathbf{X}_1, \dots, \mathbf{X}_T\} \in \mathbb{R}^{H \times W \times T}$, where each matrix $\mathbf{X}_t \in \mathbb{R}^{H \times W}$ represents t -th image frame $t = 1, \dots, T$. H , W and T denote the height, width of an image frame, and the number of image frames, respectively. The three modes of tensor $\mathcal{X}$ are height, width and time of the video.

As discussed in Section II-C, for a noisy video, it is necessary to decompose the video data $\mathcal{X}$ into the low-rank tensor $\mathcal{L}$ (the static video background), the smooth tensor $\mathcal{S}$ (the smooth moving objects in foreground), and the sparse tensor $\mathcal{E}$ (absorbing all kinds of noise), respectively. In the static background, the image frames keep unchanged along time domain. This can be achieved by restricting $\mathcal{L}$ to be a low-rank tensor in time domain. For the moving objects in the video foreground, they are continuous spatially and temporally so that they can be represented as a smooth tensor $\mathcal{S}$. But the sparse tensor $\mathcal{E}$ absorbs the random noise so that the random noise can be excluded from the background and foreground. An illustration of the video decomposition strategy for our proposed method is provided in Fig. 2. Specifically, it has the following form $\mathcal{X} = \mathcal{L} + \mathcal{S} + \mathcal{E}$ as proposed in Section I.

To model the low-rankness, the static background $\mathcal{L}$ is approximated by the well-known Tucker decomposition [11] with rank- (r_1, r_2, r_3) . Specifically, the Tucker decomposition has the following form

$$\mathcal{L} = \mathcal{G} \times_1 \mathbf{U}_1 \times_2 \mathbf{U}_2 \times_3 \mathbf{U}_3, \quad (1)$$

where $\mathbf{U}_1 \in \mathbb{R}^{H \times r_1}$, $r_1 < H$ and $\mathbf{U}_2 \in \mathbb{R}^{W \times r_2}$, $r_2 < W$ are orthogonal factor matrices for two spatial domain, $\mathbf{U}_3 \in \mathbb{R}^{T \times r_3}$, $r_3 < T$ is orthogonal factor matrix for temporal domain, core tensor $\mathcal{G} \in \mathbb{R}^{r_1 \times r_2 \times r_3}$ interacts these factors. The determination of (r_1, r_2, r_3) is provided in Section III-C. By formulating the low-rank tensor $\mathcal{L}$ using Tucker decomposition, it can reconstruct more accurate video background than the low-rank model based on matrices. Because the Tucker decomposition considers not only the spatial but also the temporal correlations in video background.

The smooth tensor $\mathcal{S}$ (moving objects) is assumed to have the spatio-temporal continuity property such that the foreground moves smoothly and coherently in the spatial and temporal directions. In the literature, imposing the spatio-temporal continuity constraints on moving objects in the foreground is well studied and proven to be effective [20], [21]. To measure the sensitivity to change of a quantity function, the derivative is often applied in mathematics. For discrete functions, difference operators are the approximation to derivative. Given a third-order tensor $\mathcal{S} \in \mathbb{R}^{H \times W \times T}$, $\mathcal{S}(x, y, t)$ indicates intensity of position (x, y) at time t , and

$$\begin{aligned} \mathcal{S}_h(x, y, t) &= \mathcal{S}_h(x+1, y, t) - \mathcal{S}_h(x, y, t), \\ \mathcal{S}_v(x, y, t) &= \mathcal{S}_v(x, y+1, t) - \mathcal{S}_v(x, y, t), \\ \mathcal{S}_t(x, y, t) &= \mathcal{S}_t(x, y, t+1) - \mathcal{S}_t(x, y, t) \end{aligned}$$

denote three difference operation results of position (x, y) at time t with periodic boundary conditions along the horizontal, vertical, and temporal directions, respectively. For simplicity of computation, all the entries of $\mathcal{S}$ can be stacked into a column vector $\mathbf{s} = \text{vec}(\mathcal{S})$, in which $\text{vec}(\cdot)$ represents the vectorization operator. $\mathbf{D}_h \mathbf{s} = \text{vec}(\mathcal{S}_h)$, $\mathbf{D}_v \mathbf{s} = \text{vec}(\mathcal{S}_v)$, and $\mathbf{D}_t \mathbf{s} = \text{vec}(\mathcal{S}_t)$ are used to denote the vectorizations of the three difference operation results, respectively, in which $\mathbf{D}_h$, $\mathbf{D}_v$, and $\mathbf{D}_t \in \mathbb{R}^{HWT \times HWT}$. Furthermore, $\mathbf{D} \mathbf{s} = [\mathbf{D}_h \mathbf{s}^\top, \mathbf{D}_v \mathbf{s}^\top, \mathbf{D}_t \mathbf{s}^\top]^\top$ is used to represent the concatenated difference operation, in which $\mathbf{D} = [\mathbf{D}_h^\top, \mathbf{D}_v^\top, \mathbf{D}_t^\top]^\top \in \mathbb{R}^{3HWT \times HWT}$. Note that the i th element in $\mathbf{D}_h \mathbf{s}$, $\mathbf{D}_v \mathbf{s}$, and $\mathbf{D}_t \mathbf{s}$ (i.e., $[\mathbf{D}_h \mathbf{s}]_i$, $[\mathbf{D}_v \mathbf{s}]_i$, and $[\mathbf{D}_t \mathbf{s}]_i$) describes the intensity changes of i th point in $\mathbf{s}$ along the horizontal, vertical, and

temporal directions, respectively. To quantify the changes of intensity, any vector norm of $[[\mathbf{D}_h \mathbf{s}]_i, [\mathbf{D}_v \mathbf{s}]_i, [\mathbf{D}_t \mathbf{s}]_i]$ can be applied. Two commonly used vector norms are ℓ_1 and ℓ_2 norms. Specifically, the anisotropic total variation norm is defined as

$$\|\mathbf{S}\|_{TV1} = \sum_i (|[\mathbf{D}_h \mathbf{s}]_i| + |[\mathbf{D}_v \mathbf{s}]_i| + |[\mathbf{D}_t \mathbf{s}]_i|) \quad (2)$$

and the isotropic total variation norm as

$$\|\mathbf{S}\|_{TV2} = \sum_i \sqrt{[\mathbf{D}_h \mathbf{s}]_i^2 + [\mathbf{D}_v \mathbf{s}]_i^2 + [\mathbf{D}_t \mathbf{s}]_i^2}, \quad (3)$$

which are the ℓ_1 and $\ell_{2,1}$ norms of $[\mathbf{D}_h \mathbf{s}, \mathbf{D}_v \mathbf{s}, \mathbf{D}_t \mathbf{s}]^\top$, respectively. The isotropic total variation of $\mathbf{S}$ is denoted by $\|\mathbf{D} \mathbf{S}\|_{2,1}$ for the sake of notations. The total variation regularization has been widely used in image and video denoising and restoration [26], [27] due to its superiority in detecting discontinuous changes in image processing.

By combing the advantages of Tucker decomposition for the low-rank tensor and total variation regularizations for the smooth tensor, the proposed smooth sparse RTD has the following formulation

$$\begin{aligned} \min_{\mathcal{G}, \mathbf{U}_j, \mathbf{S}, \mathbf{E}} \quad & \|\mathbf{E}\|_1 + \lambda \|\mathbf{S}\|_Q \\ \text{s.t.} \quad & \mathbf{X} = \mathbf{L} + \mathbf{S} + \mathbf{E}, \\ & \mathbf{L} = \mathcal{G} \times_1 \mathbf{U}_1 \times_2 \mathbf{U}_2 \times_3 \mathbf{U}_3, \end{aligned} \quad (4)$$

where the factor matrices $\mathbf{U}_j, j = 1, 2, 3$ in Tucker decomposition (1) are orthogonal in columns. The first term in the objective function measures the sparsity of the noise tensor $\mathbf{E}$ by ℓ_1 norm. The second term encodes the spatio-temporal continuity of smooth tensor $\mathbf{S}$, which is measured by the total variation regularizations defined in (2) and (3), where Q can be selected from $\{TV1, TV2\}$. $\lambda > 0$ is the coefficient to balance the two terms in the objective function. The first term in the constraints show the decomposition strategy of the tensor data and the second term in the constraints shows the tucker decomposition of the low rank tensor $\mathbf{L}$.

Remark 1 (Comparison with [20], [28]). *Similar as our paper, the work in [20], [28] also decomposes the video into three components, namely, $\mathbf{L}$, $\mathbf{S}$, and $\mathbf{E}$. However, they focus on using $\mathbf{L} + \mathbf{E}$ to model the dynamic background, where $\mathbf{L}$ represents the static background and $\mathbf{E}$ represents the small changes in the dynamic background. These two papers are all vector-based methods without using the structure information in the tensor. Besides, our paper emphasizes removing the noise in background/foreground separation. The methods in [20], [28] are also be utilized as part of the benchmarks for comparison.*

B. Optimization Algorithm

In this section, an efficient algorithm based on alternating direction method of multipliers (ADMM) [22] is proposed and implemented for solving the proposed SS-RTD model in (4). Specifically, a multi-block version of ADMM is developed. To apply ADMM, the SS-RTD model in (4) is rewritten as the

following equivalent form by introducing an auxiliary variable $\mathbf{f}$:

$$\begin{aligned} \min_{\mathcal{G}, \mathbf{U}_j, \mathbf{S}, \mathbf{f}, \mathbf{E}} \quad & \|\mathbf{E}\|_1 + \lambda \|\mathbf{f}\|_q \\ \text{s.t.} \quad & \mathbf{f} = \mathbf{D} \text{vec}(\mathbf{S}), \\ & \mathbf{X} = \mathbf{L} + \mathbf{S} + \mathbf{E}, \\ & \mathbf{L} = \mathcal{G} \times_1 \mathbf{U}_1 \times_2 \mathbf{U}_2 \times_3 \mathbf{U}_3, \end{aligned} \quad (5)$$

where the factor matrices $\mathbf{U}_j, j = 1, 2, 3$ are orthogonal in columns, and $\|\cdot\|_q$ can be selected from $\|\cdot\|_1$ or $\|\cdot\|_{2,1}$. The above optimization problem in (5) can be solved in its Lagrangian dual form by augmenting the constraints into the objective function. Specifically, the augmented Lagrangian function of problem in (5) has the follow form

$$\begin{aligned} L(\mathcal{G}, \mathbf{U}_j, \mathbf{S}, \mathbf{f}, \mathbf{E}) = & \|\mathbf{E}\|_1 + \lambda \|\mathbf{f}\|_q \\ & - \langle \boldsymbol{\lambda}^f, \mathbf{f} - \mathbf{D} \text{vec}(\mathbf{S}) \rangle + \frac{\beta^f}{2} \|\mathbf{f} - \mathbf{D} \text{vec}(\mathbf{S})\|_2^2 \\ & - \langle \boldsymbol{\Lambda}^{\mathbf{X}}, \mathbf{X} - \mathcal{G} \times_1 \mathbf{U}_1 \times_2 \mathbf{U}_2 \times_3 \mathbf{U}_3 - \mathbf{S} - \mathbf{E} \rangle \\ & + \frac{\beta^{\mathbf{X}}}{2} \|\mathbf{X} - \mathcal{G} \times_1 \mathbf{U}_1 \times_2 \mathbf{U}_2 \times_3 \mathbf{U}_3 - \mathbf{S} - \mathbf{E}\|_F^2, \end{aligned} \quad (6)$$

where $\boldsymbol{\lambda}^f, \boldsymbol{\Lambda}^{\mathbf{X}}$ are the Lagrange multiplier vector and tensor, respectively, and $\beta^f, \beta^{\mathbf{X}} > 0$ are positive penalty scalars. The optimization problem in (6) is a non-convex optimization problem. Thus, optimizing all variables simultaneously is difficult. Instead, this optimization problem (6) is solved by alternatively minimizing one variable with the others fixed. Under the framework of multi-block ADMM, the optimization problem (6) with respect to each variable can be solved by the following sub-problems.

1) *Optimization on $\mathcal{G}, \mathbf{U}_j$ or $\mathbf{L}$* : The optimization sub-problem of (6) with respect to $\mathcal{G}$ and $\mathbf{U}_j, j = 1, 2, 3$ can be rewritten as

$$\min_{\mathcal{G}, \mathbf{U}_j} \|\tilde{\mathbf{X}} - \mathcal{G} \times_1 \mathbf{U}_1 \times_2 \mathbf{U}_2 \times_3 \mathbf{U}_3\|_F^2, \text{ s.t. } \mathbf{U}_j^\top \mathbf{U}_j = \mathbf{I}, \quad (7)$$

where $\tilde{\mathbf{X}} = \mathbf{X} - \mathbf{S} - \mathbf{E} - \boldsymbol{\Lambda}^{\mathbf{X}} / \beta^{\mathbf{X}}$. The classic HOOI algorithm [11] can be applied to solve this sub-problem, which is another alternating iterative algorithm. Due to the non-convexity of the sub-problem (7), HOOI cannot obtain the optimal solution in general. However, the iterative sequence from HOOI has the monotone decreasing property.

2) *Optimization on $\mathbf{S}$* : By setting the gradient of L with respect to $\mathbf{S}$ to zero, the sub-problem of (6) with respect to $\mathbf{S}$ can be solved by the following linear equations

$$\begin{aligned} & (\beta^{\mathbf{X}} \mathbf{I} + \beta^f \mathbf{D}^* \mathbf{D}) \text{vec}(\mathbf{S}) \\ & = \beta^{\mathbf{X}} (\text{vec}(\mathbf{X}) - \text{vec}(\mathbf{L}) - \text{vec}(\mathbf{E})) - \text{vec}(\boldsymbol{\Lambda}^{\mathbf{X}}) \\ & + \mathbf{D}^* (\beta^f \mathbf{f} - \boldsymbol{\lambda}^f), \end{aligned}$$

where $\mathbf{D}^*$ indicates the adjoint of $\mathbf{D}$. Let $\mathcal{C} = \beta^{\mathbf{X}} (\mathbf{X} - \mathbf{L} - \mathbf{E}) - \boldsymbol{\Lambda}^{\mathbf{X}} + \text{ten}(\mathbf{D}^* (\beta^f \mathbf{f} - \boldsymbol{\lambda}^f))$, where $\text{ten}(\cdot)$ denotes the tensorization operator. Thanks to the block-circulant structure of the matrix corresponding to the operator $\mathbf{D}^* \mathbf{D}$, it can be diagonalized by the 3D FFT matrix [29]. Therefore, $\mathbf{S}$ can be fast computed by

$$\text{ifftn} \left(\frac{\text{fftn}(\mathcal{C})}{\beta^{\mathbf{X}} \mathbf{1} + \beta^f \Phi} \right), \quad (8)$$

where $\Phi = |\text{fft}(\mathbf{D}_h)|^2 + |\text{fft}(\mathbf{D}_v)|^2 + |\text{fft}(\mathbf{D}_t)|^2$ is not related to the data and model parameters so that it only needs to be calculated once in the whole algorithm. fft and ifft indicate fast 3D Fourier transform and its inverse transform, respectively.

3) *Optimization on $\mathbf{f}$* : The sub-problem of (6) with respect to $\mathbf{f}$ can be rewritten as

$$\min_{\mathbf{f}} \lambda \|\mathbf{f}\|_q + \frac{\beta^{\mathbf{f}}}{2} \|\mathbf{f} - (\mathbf{D}\text{vec}(\mathbf{S}) + \frac{\lambda^{\mathbf{f}}}{\beta^{\mathbf{f}}})\|_2^2. \quad (9)$$

For $q = \{1\}$ (anisotropic total variation), the well-known soft-thresholding operator [30] can be applied to solve this sub-problem as follows

$$\mathbf{f} = \text{soft}(\mathbf{D}\text{vec}(\mathbf{S}) + \frac{\lambda^{\mathbf{f}}}{\beta^{\mathbf{f}}}, \frac{\lambda}{\beta^{\mathbf{f}}}), \quad (10)$$

where the soft-thresholding operator $\text{soft}(\mathbf{A}, \tau) = \text{sign}(\mathbf{A}) \cdot \max(|\mathbf{A}| - \tau, 0)$ is performed element-wisely.

For $q = \{2, 1\}$ (isotropic total variation), denote $\mathbf{p}_h = \mathbf{D}_h \mathbf{f} + \frac{\lambda^{\mathbf{f}}}{\beta^{\mathbf{f}}}$. The definitions for $\mathbf{p}_v$ and $\mathbf{p}_t$ are analogous. $\mathbf{f}_h$ can be efficiently computed by

$$\mathbf{f}_h = \max(\mathbf{p} - \frac{\lambda}{\beta^{\mathbf{f}}}, 0) \cdot \frac{\mathbf{p}_h}{\mathbf{p}}, \quad (11)$$

where $\mathbf{p} = \max(\sqrt{|\mathbf{p}_h|^2 + |\mathbf{p}_v|^2 + |\mathbf{p}_t|^2}, \varepsilon)$ which is a element-wise operator, and $\varepsilon > 0$ is a small positive constant.

4) *Optimization on $\mathbf{E}$* : The sub-problem of (6) with respect to $\mathbf{E}$ can be solved by

$$\mathbf{E} = \text{soft}(\mathbf{X} - \mathbf{L} - \mathbf{S} - \mathbf{A}^{\mathbf{X}}/\beta^{\mathbf{X}}, \frac{2}{\beta^{\mathbf{X}}}), \quad (12)$$

where $\mathbf{L} = \mathbf{G} \times_1 \mathbf{U}_1 \times_2 \mathbf{U}_2 \times_3 \mathbf{U}_3$.

5) *Updating multipliers*: According to the ADMM, the multipliers associated with $\mathbf{L}$ are updated by the following formulas

$$\begin{aligned} \lambda^{\mathbf{f}} &\leftarrow \lambda^{\mathbf{f}} - \gamma \beta^{\mathbf{f}} (\mathbf{f} - \mathbf{D}\text{vec}(\mathbf{S})), \\ \mathbf{A}^{\mathbf{X}} &\leftarrow \mathbf{A}^{\mathbf{X}} - \gamma \beta^{\mathbf{X}} (\mathbf{X} - \mathbf{L} - \mathbf{S} - \mathbf{E}), \end{aligned} \quad (13)$$

where $\gamma > 0$ is a parameter associated with convergence rate, and the penalty parameters $\beta^{\mathbf{f}}$ and $\beta^{\mathbf{X}}$ follow an adaptive updating scheme as suggested in [31]. Take $\beta^{\mathbf{f}}$ as an example,

$$\beta^{\mathbf{f}} = \begin{cases} c_1 \beta^{\mathbf{f}}, & \text{if } \text{Err}(\mathbf{f}^{k+1}) \geq c_2 \text{Err}(\mathbf{f}^k) \\ \beta^{\mathbf{f}}, & \text{otherwise.} \end{cases} \quad (14)$$

where $\text{Err}(\mathbf{f}^k) = \|\mathbf{f}^k - \mathbf{D}\text{vec}(\mathbf{S}_k)\|$. As suggested in [21], $\gamma = 1.1$, and c_1, c_2 can be taken as 1.15 and 0.95, respectively. The proposed algorithm for SS-RTD can now be summarized in Algorithm 1. The derivations for (7), (9), and (12) are provided in the Appendix A.

C. Implementation Details and Complexity Analysis

In the SS-RTD model (4), there exist four parameters, i.e., r_1, r_2, r_3 and λ , respectively, where r_1 and r_2 control the complexity of spatial redundancy, r_3 controls the complexity of temporal redundancy, and λ handles a trade-off between noise and foreground modeling. In all experiments, r_1 and r_2 are set to values of $\text{ceil}(0.80 \times H)$ and $\text{ceil}(0.80 \times W)$,

Algorithm 1 Optimization algorithm for SS-RTD

Input: Noisy video data $\mathbf{X}$; The algorithm parameters: λ .
Initialization: $r_1 = \text{ceil}(0.8 \times H)$, $r_1 = \text{ceil}(0.8 \times W)$; $\gamma = 1.1$; $c_1 = 1.15, c_2 = 0.95$; $\mathbf{L}$ is initialized by (r_1, r_2, r_3) -Tucker decomposition of $\mathbf{X}$; $\mathbf{S} = \mathbf{X} - \mathbf{L}$; $\beta^{\mathbf{f}}, \beta^{\mathbf{X}}$ are initialized by $\frac{1e^{+1}}{\text{mean}(\mathbf{X})}, \frac{4e^{-1}}{\text{mean}(\mathbf{X})}$; Other variables are initialized by 0.
1: **while** not converged **do**
2:   Updating $\mathbf{G}$ and $\mathbf{U}_j$ (i.e., $\mathbf{L}$) via (7);
3:   Updating $\mathbf{S}$ via (8);
4:   Updating $\mathbf{f}$ via (10) for anisotropic total variation (i.e., $q = \{1\}$) or (11) for isotropic one (i.e., $q = \{2, 1\}$);
5:   Updating $\mathbf{E}$ via (12);
6:   Updating multipliers and the related parameters via (13) and (14).
7: **end while**

---

respectively, where $\text{ceil}(\cdot)$ is the operator to round the element to the nearest integer greater than or equal to that element. For r_3 , it takes the value 1 for all experiments as it has proven to be effective in [21], [32]. In terms of λ , it needs to be carefully tune based on the data. Specifically, λ is taken in the range $[0.2, 1]$, $[1, 3]$ for $q = \{1\}$ and $q = \{2, 1\}$. Empirically, the proposed algorithm can achieve satisfactory performance with any $\lambda \in [0.2, 1]$ for $q = \{1\}$. Therefore, it can be concluded that the proposed SS-RTD is a “parameter-free” algorithm.

Next, the computational complexity of the proposed SS-RTD is analyzed. In Algorithm 1, one outer iteration from line 1 to 7 includes updating rules for $\mathbf{L}, \mathbf{S}, \mathbf{f}, \mathbf{E}$, and multipliers, respectively. Updating $\mathbf{L}$ is achieved by the iterative HOOI algorithm, which needs $O(W^3 + H^3 + T^3)$ floating operations in each inner iteration. In this paper, the HOOI algorithm iterates 20 times, then updating $\mathbf{L}$ needs $O(W^3 + H^3 + T^3)$ floating operations. At each iteration of updating $\mathbf{S}$, the main computation is from the four FFTs (including three FFTs and one inverse fast Fourier transform), each is with $O(HWT \log(HWT))$ as shown in [33]. Updating $\mathbf{f}, \mathbf{E}$, and multipliers only requires element-wise addition and shrinkage operations, say $O(HWT)$. In conclusion, each outer iteration of the proposed algorithm requires $O(W^3 + H^3 + T^3 + HWT \log(HWT))$ floating point operations.

IV. NUMERICAL STUDIES

To evaluate the performance of the proposed SS-RTD, its performance demonstrations on simulated data from open-sourced video data is presented in this section. In Section IV-A, the empirical convergence and sensitivity of the proposed algorithm are illustrated. The performances of the proposed algorithm for background subtraction and foreground detection are presented in Section IV-B and IV-C, respectively. In Section IV-B and IV-C, RTPCA¹ [18], IRTPCA² [17], TTNN³ [19], TVRPCA⁴ [20], and GoDec⁵ [28] are selected

¹<https://canyilu.github.io/code/>

²<https://github.com/chenlongxiabc/IRTPCAcode>

³<https://zhaoxile.github.io/>

⁴<https://yangliang.github.io/>

⁵<https://sites.google.com/site/godecomposition/code>

as benchmarks for comparison with the proposed SS-RTD, which are state-of-the-art methods in the related area. The benchmarks have two categories:

- (1) RTPCA, IRTPCA, TTNN are the most advanced Robust Tensor PCA algorithms in the literature;
- (2) TVRPCA and GoDec are vector-based algorithms to decompose a video into three terms.

All results in this section are the average results of 20 repetitions for comparison. The codes of SS-RTD are implemented in Matlab 2019a. The CPU of the computer to conduct experiments in this paper is an Intel® Xeon® Processor E-2286M (8-cores 2.40-GHz Turbo, 16 MB).

Performance evaluation indices and parameter tuning:

For the task of background subtraction, the peak signal-to-noise ratio (PSNR) and the structural similarity index (SSIM) are used to measure the recovery accuracy. PSNR and SSIM commonly measure the similarity of two images in intensity and structure respectively. Specifically, PSNR is defined as: $PSNR = 10 \times \log_{10} \frac{255^2}{\|\mathbf{I} - \hat{\mathbf{I}}\|_F^2}$, where $\mathbf{I}$ and $\hat{\mathbf{I}}$ are respectively the original and recovered background. SSIM measures the structural similarity of two images; see [34] for details. Average PSNR and SSIM over all image frames in the video are used to evaluate recovery performance of video background. For the task of foreground detection, F-measure is applied to assess the foreground detection performance. Average F-measure over all image frames in the video is applied to evaluate the detection performance of video foreground. Therefore, 20 repetitions are sufficient to represent the performance of each method since each repetition is the average performance of multiple image frames. For these performance indices PSNR, SSIM, and F-measure, higher values indicate the better performance.

For the proposed method as well as the benchmark methods, the parameter tuning is performed by searching from 100 sets of parameters sampled by the maximin Latin hypercube design [35] such that the average PSNR and F-measure achieve the best value for background subtraction and foreground detection, respectively. In the proposed algorithm 1, λ can be selected from $[0.2, 1]$ for $q = \{1\}$ and $[1, 3]$ for $q = \{2, 1\}$, respectively.

A. Convergence and Sensitivity Analysis

1) *Convergence Analysis:* The video dataset *Candela* from SBI dataset⁶ [36] is used in this section. In total, this video dataset has 855 image frames, where the size of each grayscale image is 288×352 . For simplicity, the first 80 image frames in the sequence are used for experiments. Therefore, the tensor size is $288 \times 352 \times 80$. One image frame from *Candela* in this experiment is shown in Fig. 5a. In the video, there is a man entering and leaving room, abandoning a bag. For each image, 10% of pixels are randomly selected to set as random integers in $[0, 255]$, and the positions of the contaminated pixels are unknown (one example is shown in Fig. 5d). To evaluate the convergence of the proposed algorithm, the relative change $relChgA = \frac{\|\mathcal{A}^k - \mathcal{A}^{k-1}\|_F}{\max(1, \|\mathcal{A}^{k-1}\|_F)}$ and the relative error $relErrA = \frac{\|\mathcal{A}^k - \mathcal{A}^0\|_F}{\max(1, \|\mathcal{A}^0\|_F)}$ are applied as the assessment

indices of algorithm convergence, where $\mathcal{A}^k$ is the result in k -th iteration and $\mathcal{A}^0$ is the ground-truth. The ground truth of the static video background is provided in the first column of Fig. 6.

The curves of the relative change of the video background $\mathcal{L}$ and the video foreground $\mathcal{S}$, and the relative error of the video background $\mathcal{L}$ are shown in Fig. 3. The curve of the

Fig. 3. The empirical convergence analysis of SS-RTD algorithm with (a) isotropic TV regularization ($q = \{2, 1\}$); (b) anisotropic TV regularization ($q = \{1\}$).

relative error of the video foreground $\mathcal{S}$ is not provided since the ground-truth video foreground for real data is unknown. In the proposed algorithm 1, parameter λ for isotropic TV regularization ($q = \{2, 1\}$) and anisotropic TV regularization ($q = \{1\}$) are set to the values of 0.4 and 0.07, respectively. The convergence of the proposed algorithm for these two cases are shown in Figs. 3a and 3b, respectively. For both $q = \{2, 1\}$ and $q = \{1\}$, the relative change converges to zero when the number of iterations keeps increasing. Meanwhile, the corresponding relative error of the video background $\mathcal{L}$ gradually decreases to a stable value. The relative error of the video background $\mathcal{L}$ for the anisotropic TV regularization converges to a smaller value than the case of the isotropic TV regularization. In general, the results in this experiment show the empirical convergence of the proposed algorithm.

Fig. 4. The sensitivity analysis of SS-RTD algorithm with isotropic TV regularization ($q = \{2, 1\}$) and anisotropic TV regularization ($q = \{1\}$).

2) *Sensitivity Analysis:* Since λ is the only tuning parameter in the proposed algorithm 1, the sensitivity of the algorithm to

⁶<https://sbmi2015.na.icar.cnr.it/SBIdataset.html>

TABLE I
BACKGROUND SUBTRACTION RESULTS COMPARISON ON DIFFERENT VIDEO DATASETS WITH DIFFERENT NOISE RATIOS

Videos	Noise Ratio	Indices	RTPCA	IRTPCA	TTNN	TVRPCA	GoDec	Proposed
<i>Candela</i>	10%	PSNR	28.08	29.54	24.20	36.31	34.10	43.27
		SSIM	0.8277	0.7298	0.5604	0.8192	0.7223	0.9019
		Time	79.88	121.22	56.52	271.12	32.67	63.35
	20%	PSNR	28.59	29.87	20.99	38.04	36.34	43.22
		SSIM	0.8178	0.7054	0.3338	0.8549	0.7730	0.9022
		Time	74.59	117.16	51.17	264.69	32.99	60.00
	30%	PSNR	29.22	31.14	18.14	36.96	34.73	42.97
		SSIM	0.7842	0.6868	0.2338	0.8393	0.6786	0.9029
		Time	71.20	117.72	51.39	261.02	33.30	60.99
<i>Caviar1</i>	10%	PSNR	25.05	25.25	24.77	31.53	29.85	35.05
		SSIM	0.7343	0.7201	0.6548	0.7824	0.7379	0.8086
		Time	65.52	101.89	52.15	272.49	32.26	58.31
	20%	PSNR	26.05	26.73	21.68	31.79	30.61	33.34
		SSIM	0.7596	0.7140	0.2845	0.7747	0.7394	0.8075
		Time	61.72	102.45	49.48	259.87	32.37	57.49
	30%	PSNR	26.92	26.76	19.09	30.60	28.96	31.80
		SSIM	0.7418	0.6924	0.1714	0.7298	0.6104	0.8062
		Time	57.90	101.95	49.39	258.30	32.74	57.74
<i>Caviar2</i>	10%	PSNR	33.47	34.37	29.32	40.87	40.81	44.34
		SSIM	0.9051	0.9108	0.7691	0.8914	0.8859	0.9500
		Time	64.23	102.45	51.97	262.08	32.71	57.88
	20%	PSNR	33.77	35.82	22.38	41.00	40.46	43.72
		SSIM	0.9032	0.8966	0.4144	0.9030	0.8949	0.9488
		Time	60.85	102.45	49.50	261.64	32.83	57.21
	30%	PSNR	33.99	36.44	18.48	40.41	38.90	42.83
		SSIM	0.8928	0.8817	0.2528	0.9026	0.8293	0.9470
		Time	58.85	102.60	49.41	259.83	32.48	57.41

Note: The bold red numbers are the best performance for each case.

parameter λ is studied to further explore its effect. The same dataset with 10% noise as the one in Section IV-A1 is applied here. However, in this experiment, λ can vary from 0.01 to 3 for both isotropic ($q = \{2, 1\}$) and anisotropic ($q = \{1\}$) TV regularizations. The results of relative error of the video background $\mathcal{L}$ for both cases are plotted in Fig. 4 vs. λ .

The relative error of the proposed SS-RTD with isotropic ($q = \{2, 1\}$) TV regularization keeps decreasing when λ increases. On the other hand, the relative error of proposed SS-RTD with anisotropic ($q = \{1\}$) TV regularization drops very fast when $0 < \lambda < 0.2$ and remains very stable when $0.2 \leq \lambda < 3$. Overall, our method with $q = \{1\}$ outperforms the one with $q = \{2, 1\}$ since it has a flat area where the performance of our method with $q = \{1\}$ is fairly good for a wide range of λ . Therefore, our method with anisotropic TV regularization with any $\lambda \in [0.2, 1]$ is recommended for practitioners.

B. Background Subtraction

In this subsection, the proposed algorithm 1 is applied to background subtraction. The video dataset *Candela* in Section IV-A, *Caviar1*, and *Caviar2* from SBI dataset [36] are used for the experiments. Figs. 5a, 5b, 5c show three image frames from the three video datasets. In the dataset of *Caviar1*, there are people slowly walking along a corridor, with mild shadows. In the dataset of *Caviar2*, there are people entering and leaving a store, standing only for few frames. For both videos, the first 80 image frames are used for experiments. Thus, the tensor data size of *Caviar1* and *Caviar2* is $256 \times 384 \times 80$. The background in each video dataset is static, which is provided as the ground truth for comparison. There are people

Fig. 5. Video datasets for background subtraction: (a) *Candela*; (b) *Caviar1*; (c) *Caviar2*; (d) *Candela* with 10% noise; (e) *Caviar1* with 10% noise; (f) *Caviar2* with 10% noise.

walking in the background, which are treated as the smooth foreground. The noisy effect is simulated the same way as in [18]. Specifically, a ratio of pixels in each image are set to random integer in $[0, 255]$, and the positions of the contaminated pixels are unknown. The corresponding noisy images from the three video datasets are shown in the second row of Fig. 5. Note that the simulated noise in the section is much harder than the additive noise like Gaussian noise since the information in the pixel is completely ruined.

In this experiment, the cases of 10%, 20%, and 30% noise are studied to show the performance on background subtraction under different noise ratio. The quantitative results of all benchmark methods with different noise ratios on

Fig. 6. Visualization results of different methods on a frame of *Candela*, *Caviar1*, and *Caviar2* for background subtraction.

simulated *Candela*, *Caviar1*, and *Caviar2* are summarized in Table I in terms of PSNR, SSIM, and Computational Time, respectively. The performance of proposed SS-RTD is the best from $q = \{1\}$ and $q = \{2, 1\}$ in terms of PSNR, our method with $q = \{1\}$ always has better background subtraction performance and computation efficiency. For all cases using PSNR and SSIM, our method can achieve the best performance for all scenarios. When the noise ratio increases, our proposed SS-RTD is the most consistent one among all the benchmark methods. Specifically, our method is performing much better than RTPCA, IRTPCA, and TTNN. This demonstrates the advantage of low-rank tucker decomposition for the static background model over the tensor nuclear norm. TVRPCA and GoDec have better performance than the tensor nuclear norm based algorithms given that TVRPCA and GoDec decompose the data into three components. Nevertheless, the proposed method shows better performance than TVRPCA and GoDec due to the advantage of tensor modeling over the vector-based method. In terms of computational time, our method ranks third among all benchmarks and is very close to the second fastest algorithm, i.e., TTNN. The quantitative result shows that the proposed algorithm not only has the best performance in terms of accuracy but also runs efficiently.

To show the visualization result, the background subtraction results from the case of 10% noise on all three video datasets are demonstrated. The visualizations of the recovered video background from each video dataset for different methods are shown in Fig. 6. For the backgrounds subtracted from RTPCA, IRPCA, and TTNN, there are people remaining in the background even though the majority of the noise is removed. For the result from GoDec, there is only a little shadow of people left in the background. For the result from TVRPCA, its performance is very close to the proposed method for *Candela*, and *Caviar2*. But it performs poorly for the *Caviar1*. In general, our method produces the cleanest background for all three video datasets.

Since the quantitative results in Table I are the best performance for each algorithm from 100 sets of tuning parameters, other descriptive statistics (for example, median, quartiles, etc.) for each algorithm are unknown to us. For the case of *Candela* with 10% noise, the boxplot in Fig. 7 summarizes five-number standard (the minimum, the maximum, the sample

Fig. 7. The boxplot of background subtraction performance for different methods on 100 sets of tuning parameters. (“+” means outliers)

median, and the first and third quartiles) for each algorithm in terms of PSNR. This result shows that the proposed method is very stable for different values of λ since the variation among these five numbers is quite small. Even though TVRPCA and GoDec can achieve good performance, they are more sensitive to the tuning parameters compared with our method, especially TVRPCA. This comparison shows that the proposed SS-RTD is a kind of parameter-free algorithm since its performance for different λ is almost the same.

C. Foreground Detection

In this subsection, our algorithm 1 is applied to foreground detection. The video datasets, i.e., *Highway*, *Office*, and *Pedestrians* from CDnet dataset⁷ [37], are used for experiments. Figs. 8a, 8b, and 8c show three image frames from the three video datasets, respectively. In the dataset of *Highway*, there are vehicles moving along the highway, where the size of each grayscale image is 240×320 . In the dataset of *Office*, there is a man entering, staying, and leaving the room, where the size of each grayscale image is 240×360 . In the dataset of *Pedestrians*, there are people walking on the road, where the size of each grayscale image is 240×360 . For all the videos, the first 80 image frames are used for experiments for the

⁷<http://jacarini.dinf.usherbrooke.ca/dataset2014/>.

TABLE II
FOREGROUND DETECTION RESULTS COMPARISON ON DIFFERENT VIDEO DATASETS WITH DIFFERENT NOISE RATIOS

Videos	Noise Ratio	Indices	RTPCA	IRTPCA	TTNN	TVRPCA	GoDec	Proposed
Highway	10%	Precision	0.6988	0.5760	0.5763	0.7366	0.8284	0.8692
		Recall	0.1799	0.1771	0.1406	0.9739	0.2261	0.9681
		F-measure	0.2803	0.2645	0.2215	0.8376	0.3480	0.9158
	20%	Time	63.28	81.84	36.38	217.20	22.99	55.15
		Precision	0.7793	0.6135	0.4215	0.3849	0.6381	0.8729
		Recall	0.1158	0.1067	0.1049	0.9977	0.1207	0.9681
Office	10%	F-measure	0.1980	0.1781	0.1641	0.5524	0.1989	0.9179
		Time	57.02	82.53	33.69	221.32	23.28	56.05
	20%	Precision	0.4746	0.5921	0.4527	0.7916	0.6788	0.8580
		Recall	0.1744	0.2049	0.1099	0.6781	0.2757	0.9510
		F-measure	0.2518	0.3013	0.1756	0.7262	0.3854	0.9006
		Time	67.97	86.99	38.20	272.92	25.78	62.00
Pedestrians	10%	Precision	0.5713	0.6411	0.5169	0.7669	0.4908	0.8613
		Recall	0.1310	0.1508	0.0890	0.6697	0.1632	0.9491
		F-measure	0.2105	0.2412	0.1508	0.7097	0.2406	0.9013
	20%	Time	60.16	86.47	36.80	270.30	26.10	62.68
		Precision	0.7143	0.5261	0.5278	0.7004	0.6852	0.7444
		Recall	0.1473	0.1517	0.0968	0.8945	0.1683	0.9528
	10%	F-measure	0.2441	0.2351	0.1635	0.7837	0.2701	0.8352
		Time	44.73	52.67	23.17	118.31	15.87	39.49
	20%	Precision	0.7775	0.5447	0.5025	0.3333	0.7082	0.6690
		Recall	0.0897	0.0906	0.0511	0.9980	0.0917	0.9283
		F-measure	0.1607	0.1551	0.0927	0.4960	0.1623	0.7770
		Time	39.44	52.76	21.37	119.20	15.86	40.08

Note: The bold red numbers are the best performance for each case.

Fig. 8. Video datasets for foreground detection: (a) Highway; (b) Office; (c) Pedestrians; (d) Highway with 20% noise; (e) Office with 20% noise; (f) Pedestrians with 20% noise.

sake of computational time. The background in each video dataset is static. The binary masks for the video foreground are provided as ground truth for comparison. To simulate the noisy effect, the salt and pepper noise [20] is applied. Specifically, a ratio of pixels in each image are set to random integer from $\{0, 255\}$, and the positions of the contaminated pixels are unknown. The corresponding noisy images from the three video datasets are shown in the second row of Fig. 8.

In this experiment, the cases of 10% and 20% noise are considered to show the performance on foreground detection under different noise ratio. The quantitative results of all benchmark methods with different noise ratios on simulated Highway, Office, and Pedestrians are summarized in Table II in terms of Precision, Recall, F-measure, and Computational Time, respectively. The performance of proposed SS-RTD is

the best from $q = \{1\}$ and $q = \{2, 1\}$ in terms of F-measure, our method with $q = \{1\}$ always has better foreground detection performance and computation efficiency. In terms of precision, our method can achieve the best performance in five out of the total six scenarios. In terms of recall, the proposed SS-RTD can achieve the best performance in three out of the total six scenarios. More importantly, our method is the best for all six scenarios in terms of F-measure. When the noise ratio increases, our proposed SS-RTD is the most consistent one among all the benchmark methods. TVRPCA and GoDec have better performance than the tensor nuclear norm based algorithms given that TVRPCA and GoDec decompose the data into three components. However, our proposed method shows better performance than TVRPCA and GoDec due to the advantage of tensor modeling over the vector-based method. In terms of computational time, our method runs efficiently even though it is not the fastest one.

To show the visualization result, the foreground detection results from the case of 20% noise ratio are demonstrated. The visualizations of one frame from each video dataset for different methods are shown in Fig. 9. For the foreground masks subtracted from RTPCA, IRPCA, TTNN, and GoDec, there are lots of noise remaining in the foreground, where the moving objects are not well detected either. For the result from TVRPCA, it can remove noise from the foreground. However, its detected foreground is not as complete as our result. In summary, our method can detect the most accurate foreground among all benchmarks even though the video is noisy.

Since the quantitative results in Table II are the best performance for each algorithm from 100 sets of tuning parameters, other descriptive statistics (for example, median, quartiles, etc.) for each algorithm are still unknown to us. For the case of Highway with 20% noise, the boxplot in Fig. 10 summarizes

Fig. 9. Visualization results of different methods on a frame of *Highway*, *Office*, and *Pedestrians* for foreground detection.

five-number standard (the minimum, the maximum, the sample

Fig. 10. The boxplot of foreground detection performance for different methods on 100 sets of tuning parameters. (“+” means outliers)

median, and the first and third quartiles) for each algorithm in terms of F-measure. This result shows that our method is very stable for different values of λ since these five numbers have very small variation. Even though TVRPCA can achieve good performance, it is very sensitive to the tuning parameters compared with our method. In this case, the worst performance from our method is still much better than the best performance from all other methods.

V. APPLICATION IN ADDITIVE MANUFACTURING

In this section, the application of proposed method in melt pool detection is presented. The same benchmark methods in Section IV are applied here. In terms of metal additive manufacturing, the electron-beam melting [9] printing process using Ti-6Al-4V alloy is used. The experimental conditions for the laser are: power 190 W; spot size 100 μm ; scan speed 0.25 m/s. The X-ray imaging system is applied to monitor the printing process in order to track the melt pool information [9]. More specifically, the melt pool boundary, i.e., the solid/liquid interface, is the main interest. In terms of X-ray imaging conditions, the pixel dimension is 2 μm , the duration of each frame is 16.67 μs , the field of view is 768 μm ×1440 μm , and the frame rate is 60 kHz. One unprocessed example from the imaging system is shown in Fig. 1a. As discussed earlier,

our naked eye can hardly identify the melt pool boundary. To enhance the boundary of the melt pool, each unprocessed X-ray image is first subtracted by the X-ray image captured 6 frames earlier and then adjusted contrast to obtain the processed X-ray image to have a wide boundary, which is shown in Fig. 1b. By doing so, lots of noise are generated. Therefore, the proposed algorithm is applied to decompose the video data and remove the noise.

In this experiment, there are 100 processed X-ray image frames capturing the melt pool dynamics, where the size of each image frame is 384 × 720. Therefore, the tensor size is 384 × 720 × 100. Since the ground truth of background and foreground is unknown, visualization results are presented to show the performance of different methods. The performance of proposed SS-RTD is represented by taking $\lambda = 0.55$ and $q = \{1\}$. In Fig. 11, the background/foreground separation results from all methods are presented. In terms of background, our method can recover the porosity defects existing in the specimen, which cannot be seen from the noisy image. Besides, the shape and location of porosity defects is consistent with the ones in Fig. 1a. GoDec can also recover the porosity defects. However, the recovered background is noisy and the melt pool can still be seen in the background. For other methods, they cannot recover the background. In terms of foreground, our method and TVRPCA can remove noise from the foreground while other methods cannot. However, our detected foreground can better show the entire melt pool compared with TVRPCA. The result in this subsection shows that our algorithm can not only recover the background with porosity defects, which are hidden by the noise, but also obtain the most accurate melt pool geometry among all methods.

VI. CONCLUSION

In this article, a new smooth sparse Robust Tensor Decomposition is developed for background/foreground separation in noisy video data. The proposed SS-RTD decomposes the video data into low-rank, smooth, and sparse components, respectively. To achieve the solutions efficiently, an algorithm based on ADMM for SS-RTD is implemented. The empirical convergence experiment shows that the proposed SS-RTD can converge and run efficiently in practice. The background subtraction and foreground detection results on simulated video

Fig. 11. Visualization results of different methods on a frame of processed X-ray video.

data and X-ray data demonstrate that our method outperforms RTPCA, IRTPCA, TTNN, TVRPCA, and GoDec, which are state-of-the-art algorithms in the literature. These results also illustrate the effectiveness of Tucker decomposition for the low-rank tensor, Total Variation regularization for the smooth tensor, and noise removal model using the sparse tensor. More importantly, the proposed SS-RTD with $q = \{1\}$ can be considered as a “parameter-free” algorithm since its performance is very stable when the only tuning parameter λ takes any value from $[0.2, 1]$.

In addition, there are still some aspects of SS-RTD that deserve further investigation. First, the video background is assumed to be static in the proposed model. However, the extension to the case of dynamic background should be further investigated. Second, in the case studies, our method with $q = \{1\}$ is always better than $q = \{2, 1\}$. Therefore, finding the reason why $q = \{1\}$ is better is one of the next steps of research. Third, the empirical convergence of the proposed algorithm based on ADMM is provided in our case study. Thereafter, the theoretical convergence of the proposed algorithm needs further research.

ACKNOWLEDGMENT

Research reported in this publication was supported by the Office of Naval Research under Award Number N00014-18-1-2794, and the Department of Defense under Award Number N00014-19-1-2728.

APPENDIX A DERIVATIONS FOR ALGORITHM 1

This appendix provides the detailed derivations for Algorithm 1. Specifically, the steps to obtain (7), (9), and (12) will be shown in details.

The terms in the augmented Lagrangian function (6) related to $\mathcal{G}, \mathbf{U}_j$ are as follow

$$\begin{aligned} & \frac{\beta^{\mathcal{X}}}{2} \|\mathcal{X} - \mathcal{G} \times_1 \mathbf{U}_1 \times_2 \mathbf{U}_2 \times_3 \mathbf{U}_3 - \mathcal{S} - \mathcal{E}\|_F^2 \\ & - \langle \Lambda^{\mathcal{X}}, \mathcal{X} - \mathcal{G} \times_1 \mathbf{U}_1 \times_2 \mathbf{U}_2 \times_3 \mathbf{U}_3 - \mathcal{S} - \mathcal{E} \rangle \\ & = \|\tilde{\mathcal{X}} - \mathcal{G} \times_1 \mathbf{U}_1 \times_2 \mathbf{U}_2 \times_3 \mathbf{U}_3\|_F^2 - \|\Lambda^{\mathcal{X}}\|_F^2 / \beta^{\mathcal{X}}, \end{aligned} \quad (15)$$

where $\tilde{\mathcal{X}} = \mathcal{X} - \mathcal{S} - \mathcal{E} - \Lambda^{\mathcal{X}} / \beta^{\mathcal{X}}$. Since the second term in the right hand side of (15) is a constant, the sub-problem with respect to $\mathcal{G}, \mathbf{U}_j$ can be represented as (7).

The terms in the augmented Lagrangian function (6) related to $\mathbf{f}$ have the following representation

$$\begin{aligned} & \frac{\beta^{\mathbf{f}}}{2} \|\mathbf{f} - \mathbf{D}_{\text{vec}}(\mathcal{S})\|_2^2 - \langle \Lambda^{\mathbf{f}}, \mathbf{f} - \mathbf{D}_{\text{vec}}(\mathcal{S}) \rangle + \lambda \|\mathbf{f}\|_q \\ & = \frac{\beta^{\mathbf{f}}}{2} \|\mathbf{f} - (\mathbf{D}_{\text{vec}}(\mathcal{S}) + \frac{\Lambda^{\mathbf{f}}}{\beta^{\mathbf{f}}})\|_2^2 + \lambda \|\mathbf{f}\|_q - \|\Lambda^{\mathbf{f}}\|_2^2 / \beta^{\mathbf{f}}, \end{aligned} \quad (16)$$

where the third term in the right hand side of (16) is a constant. Therefore, the sub-problem with respect to $\mathbf{f}$ can be represented as (9).

The terms in the augmented Lagrangian function (6) related to $\mathcal{E}$ take the following form

$$\begin{aligned} & \frac{\beta^{\mathcal{X}}}{2} \|\mathcal{X} - \mathcal{L} - \mathcal{S} - \mathcal{E}\|_F^2 \\ & - \langle \Lambda^{\mathcal{X}}, \mathcal{X} - \mathcal{L} - \mathcal{S} - \mathcal{E} \rangle + \|\mathcal{E}\|_1 \\ & = \frac{\beta^{\mathcal{X}}}{2} \|\mathcal{X} - \mathcal{L} - \mathcal{S} - \Lambda^{\mathcal{X}} / \beta^{\mathcal{X}} - \mathcal{E}\|_F^2 \\ & + \|\mathcal{E}\|_1 - \|\Lambda^{\mathcal{X}}\|_F^2 / \beta^{\mathcal{X}}, \end{aligned} \quad (17)$$

where the third term in the right hand side of (17) is a constant. Thus, the equation (12) can be derived.

REFERENCES

- [1] X. Zhou, C. Yang, and W. Yu, “Moving object detection by detecting contiguous outliers in the low-rank representation,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 35, no. 3, pp. 597–610, 2012.
- [2] T. Bouwmans, S. Javed, H. Zhang, Z. Lin, and R. Otazo, “On the applications of robust pca in image and video processing,” *Proceedings of the IEEE*, vol. 106, no. 8, pp. 1427–1457, 2018.
- [3] T. Bouwmans and E. H. Zahzah, “Robust pca via principal component pursuit: A review for a comparative evaluation in video surveillance,” *Computer Vision and Image Understanding*, vol. 122, pp. 22–34, 2014.
- [4] 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,” *Computer Science Review*, vol. 23, pp. 1–71, 2017.
- [5] J. Xu, H. Li, Z. Liang, D. Zhang, and L. Zhang, “Real-world noisy image denoising: A new benchmark,” *arXiv preprint arXiv:1804.02603*, 2018.
- [6] L. Fan, F. Zhang, H. Fan, and C. Zhang, “Brief review of image denoising techniques,” *Visual Computing for Industry, Biomedicine, and Art*, vol. 2, no. 1, pp. 1–12, 2019.
- [7] Q. Guo, C. Zhao, M. Qu, L. Xiong, S. M. H. Hojjatzadeh, L. I. Escano, N. D. Parab, K. Fezzaa, T. Sun, and L. Chen, “In-situ full-field mapping of melt flow dynamics in laser metal additive manufacturing,” *Additive Manufacturing*, vol. 31, p. 100939, 2020.
- [8] M. Akbari, S. Saedodin, D. Toghraie, R. Shoja-Razavi, and F. Kowsari, “Experimental and numerical investigation of temperature distribution and melt pool geometry during pulsed laser welding of ti6al4v alloy,” *Optics & Laser Technology*, vol. 59, pp. 52–59, 2014.

- [9] C. Zhao, K. Fezzaa, R. W. Cunningham, H. Wen, F. De Carlo, L. Chen, A. D. Rollett, and T. Sun, "Real-time monitoring of laser powder bed fusion process using high-speed x-ray imaging and diffraction," *Scientific reports*, vol. 7, no. 1, pp. 1–11, 2017.
- [10] E. J. Candès, X. Li, Y. Ma, and J. Wright, "Robust principal component analysis?" *Journal of the ACM (JACM)*, vol. 58, no. 3, pp. 1–37, 2011.
- [11] T. G. Kolda and B. W. Bader, "Tensor decompositions and applications," *SIAM review*, vol. 51, no. 3, pp. 455–500, 2009.
- [12] C. J. Hillar and L.-H. Lim, "Most tensor problems are np-hard," *Journal of the ACM (JACM)*, vol. 60, no. 6, pp. 1–39, 2013.
- [13] A. Anandkumar, Y. Deng, R. Ge, and H. Mobahi, "Homotopy analysis for tensor pca," in *Conference on Learning Theory*. PMLR, 2017, pp. 79–104.
- [14] M. E. Kilmer and C. D. Martin, "Factorization strategies for third-order tensors," *Linear Algebra and its Applications*, vol. 435, no. 3, pp. 641–658, 2011.
- [15] Z. Zhang, G. Ely, S. Aeron, N. Hao, and M. Kilmer, "Novel methods for multilinear data completion and de-noising based on tensor-svd," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014, pp. 3842–3849.
- [16] C. Lu, J. Feng, Y. Chen, W. Liu, Z. Lin, and S. Yan, "Tensor robust principal component analysis: Exact recovery of corrupted low-rank tensors via convex optimization," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 5249–5257.
- [17] Y. Liu, L. Chen, and C. Zhu, "Improved robust tensor principal component analysis via low-rank core matrix," *IEEE Journal of Selected Topics in Signal Processing*, vol. 12, no. 6, pp. 1378–1389, 2018.
- [18] 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 transactions on pattern analysis and machine intelligence*, vol. 42, no. 4, pp. 925–938, 2019.
- [19] J.-H. Yang, X.-L. Zhao, T.-Y. Ji, T.-H. Ma, and T.-Z. Huang, "Low-rank tensor train for tensor robust principal component analysis," *Applied Mathematics and Computation*, vol. 367, p. 124783, 2020.
- [20] X. Cao, L. Yang, and X. Guo, "Total variation regularized rpca for irregularly moving object detection under dynamic background," *IEEE transactions on cybernetics*, vol. 46, no. 4, pp. 1014–1027, 2015.
- [21] W. Cao, Y. Wang, J. Sun, D. Meng, C. Yang, A. Cichocki, and Z. Xu, "Total variation regularized tensor rpca for background subtraction from compressive measurements," *IEEE Transactions on Image Processing*, vol. 25, no. 9, pp. 4075–4090, 2016.
- [22] S. Boyd, N. Parikh, and E. Chu, *Distributed optimization and statistical learning via the alternating direction method of multipliers*. Now Publishers Inc, 2011.
- [23] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: A survey," *ACM computing surveys (CSUR)*, vol. 41, no. 3, pp. 1–58, 2009.
- [24] H. Yan, K. Paynabar, and J. Shi, "Anomaly detection in images with smooth background via smooth-sparse decomposition," *Technometrics*, vol. 59, no. 1, pp. 102–114, 2017.
- [25] B. Shen, R. Wang, A. C. C. Law, R. Kamath, H. Choo, and Z. Kong, "Super resolution for multi-sources image stream data using smooth and sparse tensor completion and its applications in data acquisition of additive manufacturing," *Technometrics*, no. just-accepted, pp. 1–41, 2021.
- [26] Y. Wang, J. Peng, Q. Zhao, Y. Leung, X.-L. Zhao, and D. Meng, "Hyperspectral image restoration via total variation regularized low-rank tensor decomposition," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 11, no. 4, pp. 1227–1243, 2017.
- [27] H. Zhang, L. Liu, W. He, and L. Zhang, "Hyperspectral image denoising with total variation regularization and nonlocal low-rank tensor decomposition," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 58, no. 5, pp. 3071–3084, 2019.
- [28] T. Zhou and D. Tao, "Godec: Randomized low-rank & sparse matrix decomposition in noisy case," in *Proceedings of the 28th International Conference on Machine Learning, ICML 2011*, 2011.
- [29] M. K. Ng, *Iterative methods for Toeplitz systems*. Numerical Mathematics and Scie, 2004.
- [30] D. L. Donoho, "De-noising by soft-thresholding," *IEEE transactions on information theory*, vol. 41, no. 3, pp. 613–627, 1995.
- [31] S. H. Chan, R. Khoshabeh, K. B. Gibson, P. E. Gill, and T. Q. Nguyen, "An augmented lagrangian method for total variation video restoration," *IEEE Transactions on Image Processing*, vol. 20, no. 11, pp. 3097–3111, 2011.
- [32] A. Sobral, T. Bouwmans, and E.-h. Zahzah, "Lrslibrary: Low-rank and sparse tools for background modeling and subtraction in videos," *Robust Low-Rank and Sparse Matrix Decomposition: Applications in Image and Video Processing*, 2016.
- [33] M. Tao, J. Yang, and B. He, "Alternating direction algorithms for total variation deconvolution in image reconstruction," *TR0918, Department of Mathematics, Nanjing University*, 2009.
- [34] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE transactions on image processing*, vol. 13, no. 4, pp. 600–612, 2004.
- [35] V. R. Joseph and Y. Hung, "Orthogonal-maximin latin hypercube designs," *Statistica Sinica*, pp. 171–186, 2008.
- [36] L. Maddalena and A. Petrosino, "Towards benchmarking scene background initialization," in *International conference on image analysis and processing*. Springer, 2015, pp. 469–476.
- [37] Y. Wang, P.-M. Jodoin, F. Porikli, J. Konrad, Y. Benezeth, and P. Ishwar, "Cdnnet 2014: An expanded change detection benchmark dataset," in *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, 2014, pp. 387–394.

Bo Shen (S'19) received his B.S. degree in Statistics from University of Science and Technology of China, Hefei, China, in July 2017. He is currently working on his Ph.D. in Industrial and Systems Engineering at Virginia tech, Blacksburg, VA. His research interests include optimization in machine learning, data analytics in smart manufacturing.

Rakesh R. Kamath received his M.E. degree in Materials Engineering from the Indian Institute of Science, Bengaluru, India in 2016. He is currently working on his Ph.D. in Materials Science and Engineering at the University of Tennessee, Knoxville, USA. His current research interest involves using X-ray diffraction and imaging techniques to obtain fundamental insights into additive manufacturing of structural metallic alloys.

Hahn Choo received the Ph.D. degree in Metallurgical and Materials Engineering from the Illinois Institute of Technology, Chicago, IL, USA in 1998. He is currently a Professor at the Department of Materials Science and Engineering, University of Tennessee, Knoxville, USA. His research interests include physical and mechanical behavior studies of structural alloys using X-ray and neutron diffraction and imaging techniques.

Zhenyu (James) Kong (M'02) received his B.S. and M.S. degrees in mechanical engineering from Harbin Institute of Technology, Harbin, China, in 1993 and 1995, respectively, and his Ph.D. degree from the Department of Industrial and System Engineering, University of Wisconsin–Madison, Madison, WI, USA, in 2004. He is currently a professor with the Grado Department of Industrial and Systems Engineering, Virginia Tech, Blacksburg, VA, USA. His research interests include sensing and analytics for smart manufacturing, and modeling, synthesis, and diagnosis for large and complex manufacturing systems.