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

Bo Shen, Graduate Student Member, IEEE, Rakesh R. Kamath, Hahn Choo, and Zhenyu Kong, Member, IEEE

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 components 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, and the noise is modeled by the sparsity, which is enforced by the $\ell_1$ 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 $\lambda$. Based on the case studies, our method achieves satisfying performance by taking any $\lambda \in [0.2, 1]$ with anisotropic total variation regularization. With this observation, practitioners can apply the proposed method without extensive parameter tuning work. 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
$\lambda$ The balance coefficient in the proposed objective function
$\mathcal{X}$ The order three tensor in $\mathbb{R}^{H \times W \times T}$ represented by $\{X_1, \ldots, X_T\}$
$X_t$ $t$-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)
$\mathcal{X} \times_n U$ The mode-$n$ multiplication of a tensor $\mathcal{X}$ with a matrix $U$
$\Gamma$ The core tensor in Tucker decomposition
$U_1, U_2, U_3$ The factor matrices in Tucker decomposition
$F$ The auxiliary variable
$D_0, D_1, D_2$ Three vectorizations of the difference operation along the horizontal, vertical, and temporal directions
$D$ The the concatenated difference operation, i.e., $[D_0, D_1, D_2]^T$
$\| \cdot \|_F$ The Frobenius norm
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 particular, we are interested in foreground/background separation in the presence of noise. In many real-world applications, the presence of noise is a very common but challenging issue [5], [6].

One motivating example comes from metal additive manufacturing (AM) research. In metal AM, high-speed X-ray data has been applied to study the melt pool formation and evolution [7]. The geometry of melt pool can be described by the solid/liquid interface, which can be further 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 solid/liquid interface (red boundary in Fig. 1c) cannot be identified. This is because the densities of solid and liquid states of metal material are almost the same. In practice, researchers [7], [10], [11] need to apply preprocessing techniques to enhance the melt pool boundary. In this work, 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 human 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], [12], which is a modification of the widely used statistical procedure named principal component analysis (PCA). RPCA [12] decomposes the data matrix X into the sum of a low-rank matrix L and a sparse matrix S, where the low-rankness and sparsity are measured by the nuclear norm ||·||* and l1 norm ||·||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 ||·||* 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 [13]. 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 [14], [15].

Contributed by the newly developed tensor multiplication scheme on t-SVD [16], Zhang et al. [17] proposed the tensor tubal rank as well as the tensor nuclear norm for image denoising. Based on the tensor nuclear norm, Lu et al. [18] 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 L and sparse tensor E from the data tensor X, which can be represented X = L + E. 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. There are works [19], [20] that decompose the video X into three components, namely, L, S, and E. They focus on using L + E to model the dynamic background, where L represents the static background and E represents the small changes in the dynamic background. They are all vector-based methods without using the structure information in the tensor.

Deep learning-based methods for background/foreground separation have also shown promise. In [21], the authors present a background subtraction algorithm based on spatial features learned with convolutional neural networks. In [22], background estimation at each pixel is carried out by weightless neural networks designed to learn pixel color frequency during video play, and all networks share the same rule for memory retention during training. Another supervised learning-based approach is to apply a semantic segmentation model trained on a labeled dataset containing objects of interest to directly produce foreground masks for each frame of a video. In the field of semantic segmentation, deep convolutional networks are the most popular models, such
The disadvantage of such supervised methods is that they require the classes of foreground objects during training. Instead, unsupervised techniques like RPCA and RTPCA are applicable to datasets with arbitrary semantic content. While deep learning-based techniques like RPCA and RTPCA are applicable to datasets.

To achieve this objective, a smooth sparse Robust Tensor Decomposition (SS-RTD) is proposed to decompose the data tensor into a low-rank tensor (background), a smooth tensor (foreground), and a sparse tensor (noise), namely, \( \mathbf{X} = \mathbf{L} + \mathbf{S} + \mathbf{E} \). In the SS-RTD, the background is modeled by the low-rank Tucker decomposition [13]. The spatio-temporal continuity is applied to formulate the moving objects (foreground) [20], [25]. 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 and apply it to additive manufacturing applications (Figure 1).

(ii) Implement an efficient algorithm based on alternating direction method of multipliers (ADMM) [26] 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 \( \mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N} \) and its entries by \( \mathcal{X}(i_1, i_2, \cdots, i_N) \). The multi-linear Tucker rank of an \( N \)-order tensor is the tuple of the ranks of the mode-\( n \) unfoldings \( \mathbf{X}_{(n)} \in \mathbb{R}^{I_n \times (I_1 \times I_2 \times \cdots \times I_{n-1} \times I_{n+1} \times \cdots \times I_N)} \). The inner product of two same-sized tensors \( \mathcal{X} \) and \( \mathcal{Y} \) is the sum of the products of their entries, i.e., \( \langle \mathcal{X}, \mathcal{Y} \rangle = \sum_{i_1} \cdots \sum_{i_N} \mathcal{X}(i_1, \cdots, i_N) \cdot \mathcal{Y}(i_1, \cdots, i_N) \). 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., \( \langle \mathcal{X} \times_n \mathbf{U} \rangle \).

B. Related Work

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

1) Smooth Sparse Decomposition: Yan et al. [27] proposed a smooth sparse decomposition (SSD) approach to integrate smoothing images and anomaly detection into a single task. It decomposes a signal into sparse anomalous regions \( a \), and random noise \( e \), namely, \( y = \mu + a + e \). To further enhance the model, the authors proposed to use smooth spline basis \( \mathbf{B}_a \) and sparse spline basis \( \mathbf{B}_e \) to formulate the mean and anomalies. Specifically, the enhanced SSD model is as \( y = \mathbf{B} \theta + \mathbf{B}_a \theta_a + e \), where \( \theta \) is the smooth basis coefficients and \( \theta_a \) is the sparse basis coefficients, respectively. To estimate \( \theta \) and \( \theta_a \), a least square regression penalized by the two terms is as follows

\[
\begin{align*}
\min_{\theta, \theta_a} & \quad \| \mathbf{e} \|^2_F + \tau_1 \theta^T \mathbf{A} \theta + \tau_2 \| \theta_a \|_1 \\
\text{s.t.} & \quad y = \mathbf{B} \theta + \mathbf{B}_a \theta_a + e,
\end{align*}
\]

where \( \| \mathbf{e} \|^2_F \) is the fitting error; the first penalty term measures the smoothness in \( \theta \) through the roughness matrix \( \mathbf{A} \); the
second penalty term describes the sparsity in \( \theta \), and \( \tau_1, \tau_2 > 0 \) are regularization coefficients to be tuned. Along the same direction, Yan et al. extended SSD model (1) to consider spatio-temporal information in a video, where \( B = B_t \otimes B_r \) and \( B_a = B_{ta} \otimes B_{ra} \). The idea of smooth and sparse decomposition is further extended to the literature of tensor completion [28].

2) Robust Tensor PCA: As the tensor extension of the popular Robust PCA, the recent proposed RTPCA [18] aims to recover the low rank tensor \( \mathbf{L}_0 \in \mathbb{R}^{I_1 \times I_2 \times I_3} \) and sparse tensor \( \mathbf{E}_0 \in \mathbb{R}^{I_1 \times I_2 \times I_3} \) from their sum. RTPCA solves the following convex optimization problem

\[
\min_{\mathbf{L}, \mathbf{E}} \| \mathbf{L} \|_{\text{TNN}} + \rho \| \mathbf{E} \|_1, \quad \text{s.t. } \mathbf{X} = \mathbf{L} + \mathbf{E},
\]

where \( \| \cdot \|_{\text{TNN}} \) 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 [29].

A. Proposed Smooth Sparse RTD Model

Throughout this work, it is focused on the video that can be represented as a third-order tensor \( \mathbf{X} := [\mathbf{X}_1, \ldots, \mathbf{X}_T] \in \mathbb{R}^{H \times W \times T} \), where each matrix \( \mathbf{X}_t \in \mathbb{R}^{H \times W} \) represents t-th image frame \( t = 1, \ldots, T \), \( H \), \( W \) and \( T \) denote the height, width of an image frame, and the number of image frames, respectively. The three modes of tensor \( \mathbf{X} \) 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 \( \mathbf{X} \) into the low-rank tensor \( \mathbf{L} \) (the static video background), the smooth tensor \( \mathbf{S} \) (the smooth moving objects in foreground), and the sparse tensor \( \mathbf{E} \) (absorbing impulse noise), respectively. In the static background, the image frames keep unchanged along with the time domain. This can be achieved by restricting \( \mathbf{L} \) to be a low-rank tensor in the 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 \( \mathbf{S} \).

Impulse noise [32] corruption is very common in digital images. Impulse noise is always independent and uncorrelated to the image pixels and is randomly distributed over the image. For an impulse noise corrupted image, a ratio of image pixels will be noisy, and the rest of pixels will be noise-free. Therefore, the impulse noise is absorbed by the sparse tensor \( \mathbf{E} \) so that the noise can be excluded from the background and foreground. An illustration of the video decomposition strategy for our proposed model is provided in Fig. 2. Specifically, it has the following form \( \mathbf{X} = \mathbf{L} + \mathbf{S} + \mathbf{E} \) as proposed in Section I.

To model the low-rankness, the static background \( \mathbf{L} \) is approximated by the well-known Tucker decomposition [13] with rank-(\( r_1, r_2, r_3 \)). Specifically, the Tucker decomposition has the following form

\[
\mathbf{L} = \mathbf{G} \times_1 \mathbf{U}_1 \times_2 \mathbf{U}_2 \times_3 \mathbf{U}_3,
\]

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 \( \mathbf{G} \in \mathbb{R}^{r_1 \times r_2 \times r_3} \) 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], [25]. 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

\[
\mathcal{S}_h(x, y, t) = \mathcal{S}_h(x + 1, y, t) - \mathcal{S}_h(x, y, t),
\]

\[
\mathcal{S}_v(x, y, t) = \mathcal{S}_v(x, y + 1, t) - \mathcal{S}_v(x, y, t),
\]

\[
\mathcal{S}_r(x, y, t) = \mathcal{S}_r(x, y, t + 1) - \mathcal{S}_r(x, y, t),
\]

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 \( s = \text{vec}(\mathcal{S}) \), in which \( \text{vec}(\cdot) \) represents the vectorization operation. \( \mathbf{D}_h \mathbf{s} = \text{vec}(\mathcal{S}_h) \), \( \mathbf{D}_v \mathbf{s} = \text{vec}(\mathcal{S}_v) \), and \( \mathbf{D}_r \mathbf{s} = \text{vec}(\mathcal{S}_r) \) are used to denote the vectorizations of the three difference operation results, respectively, in which \( \mathbf{D}_h, \mathbf{D}_v, \) and \( \mathbf{D}_r \in \mathbb{R}^{HWT \times HWT} \). Furthermore, \( \mathbf{D} = [\mathbf{D}_h^\top, \mathbf{D}_v^\top, \mathbf{D}_r^\top]^\top \in \mathbb{R}^{3HWT \times HWT} \) is used to represent the concatenated difference operation, in which \( \mathbf{D} = [\mathbf{D}_h^\top, \mathbf{D}_v^\top, \mathbf{D}_r^\top]^\top \in \mathbb{R}^{3HWT \times HWT} \).

To quantify the changes of intensity, any vector norm of \( [\mathbf{D}_h \mathbf{s}], [\mathbf{D}_v \mathbf{s}], [\mathbf{D}_r \mathbf{s}] \) can be applied. The commonly used vector norms are \( \ell_1 \) norm. Specifically, the anisotropic total variation norm is defined as

\[
\|\mathcal{S}\|_{TV1} = \sum_i (||\mathbf{D}_h \mathbf{s}_i||_1 + ||\mathbf{D}_v \mathbf{s}_i||_1 + ||\mathbf{D}_r \mathbf{s}_i||_1),
\]

which is the \( \ell_1 \) norm of \( [\mathbf{D}_h \mathbf{s}, \mathbf{D}_v \mathbf{s}, \mathbf{D}_r \mathbf{s}]^\top \). The total variation regularization has been widely used in image and video denoising and restoration [33], [34] due to its superiority in detecting discontinuous changes in image processing.

By combining 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

\[
\begin{aligned}
\min_{\mathbf{U}_j, \mathbf{S}, \mathbf{E}} & \quad \|\mathbf{E}\|_1 + \lambda \|\mathbf{S}\|_{TV1} \\
\text{s.t.} & \quad \mathbf{X} = \mathbf{L} + \mathbf{S} + \mathbf{E}, \\
& \quad \mathbf{L} = \mathbf{G} \times_1 \mathbf{U}_1 \times_2 \mathbf{U}_2 \times_3 \mathbf{U}_3,
\end{aligned}
\]

where the factor matrices \( \mathbf{U}_j, j = 1, 2, 3 \) in Tucker decomposition (3) are orthogonal in columns. The first term in the objective function measures the sparsity of the noise tensor \( \mathbf{E} \) by \( \ell_1 \) norm. The second term encodes the spatio-temporal continuity of smooth tensor \( \mathcal{S} \), which is measures by the total variation regularizations defined in (4). \( \lambda > 0 \) 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 [35]–[37]): [35], [37] and our paper have the same concept of three components decomposition. However, the physics meaning of each component is different, which results in a totally different problem formulation and optimization algorithm. As shown in Section II-B, [35] mainly decomposes the video into the smooth background (it will dynamically change along with time), sparse anomalies, and random noise. Reference [37] decomposes a video into the varying background, the static hotspots, and the moving hotspots. Therefore, we cannot apply methods in [35], [37] to our problem. Reference [36] had the same problem setting as our paper, which is utilized as a part of the benchmarks for comparison. The difference is that they use the matrix nuclear norm to model the low-rankness. Besides, this paper is a vector-based method without using the structure information in the tensor.

B. Optimization Algorithm

In this section, an efficient algorithm based on alternating direction method of multipliers (ADMM) [26] is proposed and implemented for solving the proposed SS-RTD model in (5). Specifically, a multi-block version of ADMM is developed. To apply ADMM, the SS-RTD model in (5) is rewritten as the following equivalent form by introducing an auxiliary variable \( f \):

\[
\begin{aligned}
\min_{\mathbf{G}, \mathbf{U}_j, \mathbf{S}, \mathbf{f}, \mathbf{E}} & \quad \|\mathbf{E}\|_1 + \lambda \|\mathbf{f}\|_1 \\
\text{s.t.} & \quad \mathbf{f} = \mathbf{D} \text{vec}(\mathbf{S}),
\end{aligned}
\]
where the factor matrices $U_j, j = 1, 2, 3$ are orthogonal in columns. The above optimization problem in (6) can be solved in its Lagrangian dual form by augmenting the constraints into the objective function. Specifically, the augmented Lagrangian function of problem in (6) has the following form

$$L(G, U_j, S_j, F, E)$$

$$= ||E||_1 + \lambda \|f\|_1$$
$$- \langle \lambda_f, f - Dvec(S) \rangle + \frac{\beta_f}{2} \|f - Dvec(S)\|_2^2$$
$$- \langle \lambda_x, X - G \times_1 U_1 \times_2 U_2 \times_3 U_3 - S - E \rangle$$
$$+ \frac{\beta_x}{2} \|X - G \times_1 U_1 \times_2 U_2 \times_3 U_3 - S - E\|_F^2,$$  \hspace{1cm} (7)

where $\lambda_f, \lambda_x$ are the Lagrange multiplier vector and tensor, respectively, and $\beta_f, \beta_x > 0$ are positive penalty scalars.

The optimization problem in (7) is a non-convex optimization problem. Thus, optimizing all variables simultaneously is difficult. Instead, this optimization problem (7) is solved by alternatively minimizing one variable with the others fixed. Under the framework of multi-block ADMM, the optimization problem (7) with respect to each variable can be solved by the following sub-problems.

1) Optimization on $G, U_j$ or $L$: The optimization sub-problem of (7) with respect to $G$ and $U_j, j = 1, 2, 3$ can be rewritten as

$$\min_{G, U_j} ||\tilde{X} - G \times_1 U_1 \times_2 U_2 \times_3 U_3||_F^2, \quad \text{s.t.} \quad U_j U_j^T = I,$$ \hspace{1cm} (8)

where $\tilde{X} = X - S - E - \lambda_x / \beta_x$. The classic HOOI algorithm [13] can be applied to solve this sub-problem, which is another alternating iterative algorithm. Due to the non-convexity of the sub-problem (8), HOOI cannot obtain the optimal solution in general. However, the iterative sequence from HOOI has the monotone decreasing property.

2) Optimization on $S$: By setting the gradient of $L$ with respect to $S$ to zero, the sub-problem of (7) with respect to $S$ can be solved by the following linear equations

$$(\beta_x X + \beta D^* D)vec(S)$$
$$= \beta_x (vec(X) - vec(L) - vec(E)) - vec(\lambda_x) + D^*(\beta f - \lambda_f),$$

where $D^*$ indicates the adjoint of $D$. Let $C = \beta_x (X - L - E) - \lambda_x + vec(D^*(\beta f - \lambda_f))$, then $\text{ten}(\cdot)$ denotes the tensorization operator. Thanks to the block-circulant structure of the matrix corresponding to the operator $D^* D$, it can be diagonalized by the 3D FFT matrix [38]. Therefore, $S$ can be fast computed by

$$\text{ifftn} \left( \frac{\text{fft}(C)}{\beta_x X + \beta_f \Phi} \right),$$ \hspace{1cm} (9)

where $\Phi = |\text{fft}(D_i)|^2 + |\text{fft}(D_0)|^2 + |\text{fft}(D_0)|^2$ is not related to the data and model parameters so that it only needs to be calculated once in the whole algorithm. $\text{fft}$ and $\text{ifft}$ indicate fast 3D Fourier transform and its inverse transform, respectively.

3) Optimization on $f$: The sub-problem of (7) with respect to $f$ can be rewritten as

$$\min_f \lambda ||f||_q + \frac{\beta_f}{2} ||f - Dvec(S) + \frac{\lambda_f}{\beta_f}||_2^2.$$ \hspace{1cm} (10)

For anisotropic total variation, the well-known soft-thresholding operator [39] can be applied to solve this sub-problem as follows

$$f = \text{soft}(Dvec(S) + \frac{\lambda_f}{\beta_f}, \lambda_f),$$ \hspace{1cm} (11)

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

4) Optimization on $E$: The sub-problem of (7) with respect to $E$ can be solved by

$$E = \text{soft}(X - L - S - \lambda_x / \beta_x, \frac{\lambda}{\beta_x}),$$ \hspace{1cm} (12)

where $\lambda > 0$ is a parameter associated with convergence rate, and the penalty parameters $\beta_f$ and $\beta_x$ follow an adaptive updating scheme as suggested in [40]. Take $\beta_f$ as an example,

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

where $\text{Err}(f^k) = ||f^k - Dvec(S_i)||$. As suggested in [25], $\gamma = 1.1$, 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 (8), (10), and (12) are provided in the Appendix.

**Algorithm 1 Optimization Algorithm for SS-RTD**

**Input:** Noisy video data $X$; The algorithm parameters: $\lambda$.

**Initialization:** $r_1 = \text{ceil}(0.8 \times |H|), r_2 = \text{ceil}(0.8 \times |W|), r_3 = 1; \gamma = 1.1; c_1 = 1.15, c_2 = 0.95; L$ is initialized by $(r_1, r_2, r_3)$-Tucker decomposition of $X$; $S = X - L; \beta_f, \beta_x$ are initialized by $\frac{1}{\text{mean}(X)}$; Other variables are initialized by $0$.

1: while $||E_{k-1}^i - E_{k-1}^{i-1}||_F > 10^{-6}$ & iter $\leq 100$ do
2: Updating $G$ and $U_j$ (i.e., $L$) via (8);
3: Updating $S$ via (9);
4: Updating $f$ via (11) for anisotropic total variation;
5: Updating $E$ via (12);
6: Updating multipliers and the related parameters via (13) and (14), iter $= \text{iter} + 1$.
7: end while
C. Implementation Details and Complexity Analysis

In the SS-RTD model (5), there exist four parameters, i.e., \( r_1, r_2, r_3 \) and \( \lambda \), respectively, where \( r_1 \) and \( r_2 \) control the complexity of spatial redundancy, \( r_3 \) controls the complexity of temporal redundancy, and \( \lambda \) 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 \( \text{ceil}(0.80 \times W) \), respectively, where \( \text{ceil}(\cdot) \) is the operator to round the element to the nearest integer greater than or equal to that element.

By doing so, the accumulation energy ratio of top normalized singular values (AccEgyR) ) attains a ratio over 0.9 for various natural images, as reported in [25]. For \( r_3 \), it takes the value 1 for all experiments so that each image frame in \( L \) is the same [41]. In terms of \( \lambda \), it needs to be carefully tuned based on the data. Specifically, \( \lambda \) is taken in the range \([0.2, 1]\). Empirically, the proposed algorithm can achieve satisfactory performance with any \( \lambda \in [0.2, 1] \). Therefore, practitioners can apply the proposed method without extensive parameter tuning.

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 \( L, S, f, E \), and multipliers, respectively. Updating \( L \) 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 \( L \) needs \( O(W^3 + H^3 + T^3) \) floating operations. At each iteration of updating \( S \), the main computation is from the four FFTs (including three FFTs and one inverse fast Fourier transform), each with \( O(H W T \log(H W T)) \) as shown in [42]. Updating \( f, E \), and multipliers only requires element-wise addition and shrinkage operations, say \( O(H W T) \). In conclusion, each outer iteration of the proposed algorithm requires \( O(W^3 + H^3 + T^3 + H W T \log(H W T)) \) floating point operations.

IV. Numerical Studies

To evaluate the performance of the proposed SS-RTD, its performance on 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 [31], IRTPCA2 [29], TTNN3 [30], TVRPCA4 [20], GoDec5 [19], and PRPCA6 [36] are selected 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, and TTNN are the most advanced Robust Tensor PCA algorithms in the literature;
2) TVRPCA, GoDec, and PRPCA are vector-based algorithms to decompose a video into three components.

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:

\[
\text{PSNR} = 10 \times \log_{10} \frac{255^2}{\| \hat{I} - I \|_2^2},
\]

where \( I \) and \( \hat{I} \) are respectively the original and recovered background. SSIM measures the structural similarity of two images; see [43] 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 [44] such that the average PSNR and F-measure achieve the best value for background subtraction and foreground detection, respectively. In the proposed algorithm 1, \( \lambda \) can be selected from \([0.2, 1]\) based on the empirical results in Section IV-A.

A. Convergence and Sensitivity Analysis

1) Convergence Analysis: The video dataset Candela from SBI dataset7 [45] is used in this section. In total, this video dataset has 855 image frames, where the size of each grayscale image is \( 288 \times 352 \). 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

\[
\text{relChgA} = \frac{\| \mathbf{A}^t - \mathbf{A}^{t-1} \|_F}{\max(1, \| \mathbf{A}^{t-1} \|_F)}
\]

and the relative error \( \text{relErrA} = \frac{\| \mathbf{A}^t - \mathbf{A}^0 \|_F}{\max(1, \| \mathbf{A}^0 \|_F)} \) are applied as the assessment indices of algorithm convergence, where \( \mathbf{A}^t \) is the result in \( k \)-th iteration and \( \mathbf{A}^0 \) 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 \( L \) and the video foreground \( S \), and the relative error of the

7https://sbmi2015.na.icar.cnr.it/SBIdataset.html
video background $L$ are shown in Fig. 3. The curve of the relative error of the video foreground $S$ is not provided since the ground-truth video foreground for real data is unknown. In the proposed algorithm 1, parameter $\lambda$ is set to the values of 0.4. From Fig. 3, the relative change converges to zero when the number of iterations keeps increasing. Meanwhile, the corresponding relative error of the video background $L$ gradually decreases to a stable value. In general, the results in this experiment show the empirical convergence of the proposed algorithm.

2) Sensitivity Analysis: Since $\lambda$ is the only tuning parameter in the proposed algorithm 1, the sensitivity of the algorithm to parameter $\lambda$ is studied to further explore its effect. The same dataset with 10% noise as in Section IV-A1 is applied here. However, in this experiment, $\lambda$ can vary from 0.01 to 3. The results of relative error of the video background $L$ for both cases are plotted in Fig. 4 vs. $\lambda$. The relative error of proposed SS-RTD drops very fast when $0 < \lambda < 0.2$ and remains very stable when $0.2 \leq \lambda < 3$. Overall, our method has a flat area where the performance of our method is fairly good for a wide range of $\lambda$. Therefore, our method 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 [45] 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 walking in the background, which are treated as the smooth foreground. The noisy effect is simulated the same way as in [31]. 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 simulated Candela, Caviar1, and Caviar2 are summarized in Table I regarding PSNR, SSIM, and Computational Time, respectively. 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 approach performs much better than RTPCA, IRTPCA, and TTNN. This result demonstrates the advantage of low-rank Tucker decomposition for the static background model over the tensor nuclear norm. PRPCA shows similar performance with the nuclear norm-based algorithms, while TVRPCA and GoDec have better performance than PRPCA and these nuclear norm-based algorithms. Regarding PSNR and SSIM, our proposed method achieves the best performance due to the advantage of tensor modeling over the vector-based method. Our approach ranks third among all benchmarks in terms of computational time, but it is within the same level of magnitude as the fastest one (GoDec) and very close to the second-fastest algorithm, i.e., TTNN. PRPCA is the most time-consuming one. These results demonstrate that the proposed
algorithm has the best performance in terms of accuracy and 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. Our proposed approach generally produces the cleanest background for all three video datasets. For the backgrounds generated using RTPCA, IRPCA, and TTNN, people still remain in the background even though most of the noise is removed. For the results from TVRPCA and GoDec, their performance is very close to the proposed method for Candela, and Caviar2. But they perform poorly for the Caviar1, where there is shadow of people left in the background. PRPCA can almost recover the background without people, but the blurriness is high.

Since the quantitative results in Table I are the best performance for each algorithm from 100 sets of tuning parameters, boxplots are used to show the performance variability for each algorithm. For the case of 10% noise, the boxplots for Candela, Caviar1, and Caviar2 in Fig. 7 summarize five-number standards (the minimum, the maximum, the sample 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 $\lambda$ 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 than our method, especially TVRPCA. PRPCA is a very stable algorithm. However, its performance is inferior. This comparison shows that the proposed SS-RTD is a ready-to-use algorithm since its performance is robust with different $\lambda$.

C. Foreground Detection

In this subsection, our algorithm 1 is applied to foreground detection. The video datasets, i.e., Highway, Office,
TABLE II
FOREGROUND DETECTION RESULTS COMPARISON ON DIFFERENT VIDEO DATASETS WITH DIFFERENT NOISE RATIOS

| Videos       | Noise Ratio | Indices     | RTPCA | IRTPCA | TTNn | TVRPCA | GoDec | PRCPA | Mask | R-CNN | DeepLab | Proposed |
|--------------|-------------|-------------|-------|--------|------|--------|-------|-------|------|-------|---------|----------|
| Highway      | 10%         | Precision  | 0.6983 | 0.5760 | 0.5763 | 0.7365 | 0.8284 | 0.3610 | 0.0600 | 0.2166 | 0.8892   |
|              |             | Recall     | 0.1799 | 0.1771 | 0.1406 | 0.9739 | 0.2261 | 0.4527 | NaN   | 0.0350 | 0.9681   |
|              |             | F-measure  | 0.2803 | 0.2645 | 0.2215 | 0.8376 | 0.3480 | 0.1554 | 0.0890 | 0.0565 | 0.9158   |
|              |             | Time (Sec) | 63.28  | 81.84  | 36.38  | 217.20 | 22.99  | 407.25 | 55.15 |        |         |
|              | 20%         | Precision  | 0.7793 | 0.6135 | 0.4215 | 0.3849 | 0.6381 | 0.3665 | 0.0000 | 0.0024 | 0.8729   |
|              |             | Recall     | 0.1158 | 0.1067 | 0.1049 | 0.9977 | 0.1207 | 0.3086 | NaN   | NaN    | 0.9681   |
|              |             | F-measure  | 0.1980 | 0.1781 | 0.1641 | 0.5524 | 0.1989 | 0.1439 | 0.0000 | 0.0044 | 0.9179   |
|              |             | Time (Sec) | 57.02  | 82.93  | 33.69  | 221.32 | 23.28  | 407.22 | 56.05 |        |         |
| Office       | 10%         | Precision  | 0.4746 | 0.5921 | 0.4527 | 0.7916 | 0.6788 | 0.8314 | 0.4878 | 0.4473 | 0.8580   |
|              |             | Recall     | 0.1744 | 0.2409 | 0.1099 | 0.6781 | 0.2757 | 0.0859 | 0.9289 | 0.1532 | 0.9510   |
|              |             | F-measure  | 0.2518 | 0.3013 | 0.1756 | 0.7262 | 0.3854 | 0.1320 | 0.5334 | 0.2146 | 0.9006   |
|              |             | Time (Sec) | 67.97  | 86.99  | 38.20  | 272.92 | 25.78  | 496.91 | 62.00 |        |         |
|              | 20%         | Precision  | 0.5713 | 0.6411 | 0.5109 | 0.7669 | 0.4908 | 0.7495 | 0.2911 | 0.0346 | 0.8613   |
|              |             | Recall     | 0.1310 | 0.1508 | 0.0890 | 0.6697 | 0.1632 | 0.0992 | 0.9481 | 0.3979 | 0.9491   |
|              |             | F-measure  | 0.2105 | 0.2412 | 0.1508 | 0.7097 | 0.2406 | 0.1306 | 0.3369 | 0.0499 | 0.9013   |
|              |             | Time (Sec) | 60.16  | 86.47  | 36.80  | 270.30 | 26.10  | 498.38 | 62.58 |        |         |
| Pedestrians  | 10%         | Precision  | 0.7143 | 0.5261 | 0.5278 | 0.7004 | 0.6852 | 0.4095 | 0.0000 | 0.2787 | 0.7444   |
|              |             | Recall     | 0.1473 | 0.1517 | 0.0968 | 0.8945 | 0.1683 | 0.0916 | 0.0338 | NaN    | 0.9528   |
|              |             | F-measure  | 0.2441 | 0.2351 | 0.1635 | 0.7837 | 0.2701 | 0.0651 | 0.0000 | 0.0598 | 0.8352   |
|              |             | Time (Sec) | 44.73  | 52.67  | 23.17  | 118.31 | 15.87  | 308.20 |        |        | 39.49    |
|              | 20%         | Precision  | 0.7775 | 0.5447 | 0.5025 | 0.3333 | 0.7082 | 0.4041 | 0.1164 | 0.0000 | 0.6690   |
|              |             | Recall     | 0.0897 | 0.0906 | 0.0511 | 0.9980 | 0.0917 | 0.0546 | 0.2983 | 0.0296 | 0.9283   |
|              |             | F-measure  | 0.1607 | 0.1551 | 0.0927 | 0.4960 | 0.1623 | 0.0583 | 0.0611 | 0.0001 | 0.7770   |
|              |             | Time (Sec) | 39.44  | 52.76  | 21.37  | 119.20 | 15.86  | 307.93 | 40.08 |        |         |

Note: The bold red numbers are the best performance for each case; "__" means not applicable since they are pretrained models.

and Pedestrians from CDNet dataset$^6$ [46], 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 $240 \times 320$.

In the dataset of Office, there is a man entering, staying, and leaving the room, where the size of each grayscale image is $240 \times 360$. In the dataset of Pedestrians, there are people walking on the road, where the size of each grayscale image is $240 \times 360$. For all the videos, the first 80 image frames are used for experiments for the 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 the experiment, we apply

![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.](image-url)
two pretrained deep learning models\(^9\) on our datasets directly: a Mask R-CNN [23] model trained on the COCO 2014 dataset and a DeepLab [24] model trained on the PASCAL VOC dataset.

In this experiment, the cases of 10% and 20% noise are investigated to show the performance on foreground detection under different noise ratios. The quantitative results of all benchmark methods with different noise ratios of simulated Highway, Office, and Pedestrians are summarized in Table II in terms of Precision, Recall, F-measure, and Computational Time, respectively. 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 five 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 and PRPCA. However, our proposed approach shows best performance due to the advantage of tensor modeling over the vector-based method. The performances of two deep learning model, i.e., Mask R-CNN and DeepLab, are very inferior since they are not designed for noisy videos. Our method runs efficiently in computational time, even though it is not the fastest. PRPCA is still the slowest one.

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

Since the quantitative results in Table II are the best performance from all other methods, boxplots are used to show the performance variability for each algorithm. For the case of 20% noise, the boxplots for Highway, Office, and Pedestrians in Fig. 10 summarize five-number standards (the minimum, the maximum, the sample 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 $\lambda$ since these five numbers have very small variations. Even though TVRPCA can achieve good performance, it is very sensitive to the tuning parameters compared with our method. PRPCA is a very stable algorithm. However, its performance is poor. In this experiment, 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. The laser powder bed fusion (LPBF) process using Ti-6Al-4V alloy starts from the following specimen shown in Fig. 11a. The laser is applied on the top surface, which moves from the bottom left to the top right. The experimental conditions for the laser are: power 190 W; spot size 100 $\mu$m; scan speed 0.25 m/s. Although the process is conceptually simple, there are many highly dynamic and transient physical phenomena involved because of the extremely high heating and cooling rates, e.g., melting and partial vaporization of metallic powders, the flow of the molten metal, and rapid solidification, etc. To in-situ probe the dynamics of the LPBF process, a high-speed X-ray imaging system at the Advanced Photon Source (APS), as shown in Fig. 11b is applied, where an X-ray beam goes through the thickness of the sample as shown in Fig. 11a. In terms of X-ray imaging conditions, the pixel dimension is 2 $\mu$m, the duration of each frame is 16.67 $\mu$s, the field of view is $768\mu$m×$1440\mu$m, and the frame rate is 60 kHz. The melt pool boundary, i.e., the solid/liquid interface, can be extracted from the high-speed X-ray images to track the melt pool information. The melt pool boundary can be further used to characterize the solidification rate [9], which plays a critical role in determining the microstructure in additively manufactured metals. One unprocessed example from the imaging system is shown in Fig. 1a. As discussed earlier, it is hard to 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 six frames earlier and then adjusted contrast to obtain the processed X-ray image to have a wide boundary, which is shown in Fig. 1b. The number six is determined by trying various numbers, where six gives the best visualization. By doing so, lots of noise are generated. In the literature, researchers [7], [10], [11] manually extract the melt pool boundary from the processed X-ray image. In this work, the proposed algorithm is the first attempt to extract the melt pool information by decomposing the video data and removing 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 $384 \times 720$. Therefore, the tensor size is $384 \times 720 \times 100$. 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 $\lambda = 0.55$. In Fig. 12, 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. PRPCA can recover a noiseless background while the porosity defects cannot be identified. 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 section shows

\(^9\)https://github.com/matlab-deep-learning
that our algorithm can recover the background with porosity
defects, which are hidden by the noise, and obtain the most
accurate melt pool geometry among all methods.

VI. CONCLUSION

In this article, a new smooth sparse Robust Tensor Decom-
position 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 convergence experiment shows that the proposed SS-RTD decomposes the video data into low-rank, smooth, and sparse components, in noisy video data. The proposed SS-RTD can be considered as a ready-to-use algorithm since its performance is very stable when the only tuning parameter $\lambda$ 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, 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.

**APPENDIX**

**DERIVATIONS FOR ALGORITHM 1**

This appendix provides the detailed derivations for Algorithm 1. Specifically, the steps to obtain (8), (10), and (12) will be shown in details.

The terms in the augmented Lagrangian function (7) related to $\mathcal{G}, \mathbf{U}_j$ are as follow

\[
\begin{align*}
\frac{\beta}{2} & \|\mathbf{X} - \mathcal{G} \times_1 \mathbf{U}_1 \times_2 \mathbf{U}_2 \times_3 \mathbf{U}_3 - \mathbf{S} - \mathbf{E}\|_{F}^2 \\
& - \langle \Lambda^{X}, \mathbf{X} - \mathcal{G} \times_1 \mathbf{U}_1 \times_2 \mathbf{U}_2 \times_3 \mathbf{U}_3 - \mathbf{S} - \mathbf{E} \rangle \\
& = \|\mathbf{X} - \mathcal{G} \times_1 \mathbf{U}_1 \times_2 \mathbf{U}_2 \times_3 \mathbf{U}_3\|_{F}^2 - \|\Lambda^{X}\|_{F}^2 / \beta \mathbf{X},
\end{align*}
\]

(15)

where $\mathbf{X} = \mathbf{S} - \mathbf{E} - \Lambda^{X} / \beta \mathbf{X}$ 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 (8).

The terms in the augmented Lagrangian function (7) related to $\mathbf{f}$ have the following representation

\[
\begin{align*}
\frac{\beta f}{2} & \|\mathbf{f} - \mathbf{D} \text{vec}(\mathbf{S})\|_2^2 - \langle \lambda f, \mathbf{f} - \mathbf{D} \text{vec}(\mathbf{S}) \rangle + \lambda \|\mathbf{f}\|_q \\
& = \frac{\beta f}{2} \|\mathbf{f} - \mathbf{D} \text{vec}(\mathbf{S}) + \lambda f / \beta f\|_2^2 + \lambda \|\mathbf{f}\|_q - \|\lambda f\|_2^2 / \beta f,
\end{align*}
\]

(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 (10).

**REFERENCES**

[1] X. Zhou, C. Yang, and W. Yu, “Moving object detection by detecting contiguous outliers in the low-rank representation,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 3, pp. 597–610, Mar. 2013.

[2] 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.

[3] T. Bouwmans and E. H. Zahzah, “Robust PCA via principal component pursuit: A review for a comparative evaluation in video surveillance,” *Comput. Vis. Image Understanding.*, vol. 122, pp. 22–34, May 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,” *Comput. Sci. Rev.*, vol. 23, pp. 1–71, Jan. 2017.

[5] J. Xu, H. Li, Z. Liang, D. Zhang, and L. Zhang, “Real-world noisy image denoising: A new benchmark,” 2018, arXiv:1804.02603.

[6] L. Fan, F. Zhang, H. Fan, and C. Zhang, “Brief review of image denoising techniques,” *Vis. Comput. Ind., Biomed., Art.*, vol. 2, no. 1, pp. 1–12, Dec. 2019.

[7] Q. Guo et al., “In-situ full-field mapping of melt flow dynamics in laser metal additive manufacturing,” *Additive Manuf.*, vol. 31, Jan. 2020, Art. no. 100939.

[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,” *Opt. Laser Technol.*, vol. 59, pp. 52–59, Jul. 2014.

[9] C. Zhao et al., “Real-time monitoring of laser powder bed fusion process using high-speed X-ray imaging and diffraction,” *Sci. Rep.*, vol. 7, no. 1, pp. 1–11, Dec. 2017.

[10] S. J. Wolff et al., “In-situ observations of directed energy deposition additive manufacturing using high-speed X-ray imaging,” *JOM*, vol. 73, no. 1, pp. 189–200, Jan. 2021.

[11] J. Gillespie et al., “In situ characterization of laser-generated melt pools using synchronized ultrasound and high-speed X-ray imaging,” *J. Acoust. Soc. Amer.*, vol. 150, no. 4, pp. 2409–2420, Oct. 2021.

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

[13] T. G. Kolda and B. W. Bader, “Tensor decompositions and applications,” *SIAM Rev.*, vol. 51, no. 3, pp. 455–500, 2009.

[14] C. J. Hillar and L.-H. Lim, “Most tensor problems are NP-hard,” *J. ACM*, vol. 60, no. 6, pp. 1–39, Nov. 2013.

[15] A. Anandkumar, Y. Deng, R. Ge, and H. Mobahi, “Homotopy analysis for tensor PCA,” in *Proc. Conf. Learn. Theory*, 2017, pp. 79–104.

[16] M. E. Kilmer and C. D. Martin, “Factorization strategies for third-order tensors in the right hand side of (16) is a constant. Therefore, the sub-problem with respect to $\mathbf{f}$ can be represented as (10).

The terms in the augmented Lagrangian function (7) related to $\mathbf{E}$ take the following form

\[
\begin{align*}
\frac{\beta}{2} & \|\mathbf{X} - \mathbf{L} - \mathbf{S} - \mathbf{E}\|_{F}^2 \\
& - \langle \Lambda^{X}, \mathbf{X} - \mathbf{L} - \mathbf{S} - \mathbf{E} \rangle + \|\mathbf{E}\|_1 \\
& = \frac{\beta}{2} \|\mathbf{X} - \mathbf{L} - \mathbf{S} - \Lambda^{X}/\beta \mathbf{X} - \mathbf{E}\|_{F}^2 \\
& + \|\mathbf{E}\|_1 - \|\Lambda^{X}\|_{F}^2 / \beta \mathbf{X},
\end{align*}
\]

(17)

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

[17] 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 *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 3842–3849.

[18] 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 *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 5249–5257.

[19] T. Tao and D. Tao, “G0Dec: Randomized low-rank & sparse matrix decomposition in noisy case,” in *Proc. 28th Int. Conf. Mach. Learn. (ICML)*, 2011, pp. 33–40.
[20] X. Cao, L. Yang, and X. Guo, “Total variation regularized RPCA for irregularly moving object detection under dynamic background,” IEEE Trans. Cybern., vol. 46, no. 4, pp. 1014–1027, Apr. 2016.

[21] M. Braham and M. Van Droogenbroeck, “Deep background subtraction with scene-specific convolutional neural networks,” in Proc. Int. Conf. Syst., Signals Image Process. (IWSSIP), May 2016, pp. 1–4.

[22] M. De Gregorio and M. Giordano, “Background estimation by weight-less neural networks,” Pattern Recognit. Lett., vol. 96, pp. 55–65, Sep. 2017.

[23] K. He, G. Gkioxari, P. Dollár, and R. Girshick, “Mask R-CNN,” in Proc. IEEE Int. Conf. Comput. Vis., Oct. 2017, pp. 2961–2969.

[24] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, “Encoder-decoder with atrous separable convolution for semantic image segmentation,” in Proc. Eur. Conf. Comput. Vis. (ECCV), 2018, pp. 801–818.

[25] W. Cao et al., “Total variation regularized tensor RPCA for background subtraction from compressive measurements,” IEEE Trans. Image Process., vol. 25, no. 9, pp. 4075–4090, Sep. 2016.

[26] S. Boyd, N. Parikh, and E. Chu, Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers. New York, NY, USA: Now, 2011.

[27] H. Yan, K. Paynabar, and J. Shi, “Anomaly detection in images with smooth background via smooth-sparse decomposition,” Technometrics, vol. 60, no. 1, pp. 102–114, 2017.

[28] 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, vol. 64, no. 1, pp. 2–17, 2021.

[29] 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, 2018.

[30] 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,” Appl. Math. Comput., vol. 367, Feb. 2020, art. no. 124783.

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

[32] S. Indu and C. Ramesh, “A noise fading technique for images highly corrupted with impulse noise,” in Proc. Int. Conf. Computing: Theory Appl. (ICCTA), Mar. 2007, pp. 627–632.

[33] Y. Wang, J. Peng, Q. Zhao, D. Meng, Y. Leung, and X.-L. Zhao, “Hyperspectral image restoration via total variation regularized low-rank tensor decomposition,” IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens., vol. 11, no. 4, pp. 1227–1243, Apr. 2018.

[34] H. Zhang, L. Liu, W. He, and L. Zhang, “Hyperspectral image denoising with total variation regularization and nonlocal low-rank tensor decomposition,” IEEE Trans. Geosci. Remote Sens., vol. 58, no. 5, pp. 3071–3084, May 2019.

[35] H. Yan, K. Paynabar, and J. Shi, “Real-time monitoring of high-dimensional functional data streams via spatio-temporal smooth sparse decomposition,” Technometrics, vol. 60, no. 2, pp. 181–197, 2018.

[36] B. E. Moore, C. Gao, and R. R. Nadakuditi, “Panoramic robust PCA for foreground-background separation on noisy, free-motion camera video,” IEEE Trans. Comput. Imag., vol. 5, no. 2, pp. 195–211, Jun. 2019.

[37] S. Mou, A. Wang, C. Zhang, and J. Shi, “Additive tensor decomposition considering structural data information,” IEEE Trans. Autom. Sci. Eng., early access, Jul. 26, 2021, doi: 10.1109/TASE.2021.3096964.

[38] M. K. Ng, “Iterative methods for Toeplitz systems,” in Numerical Mathematics and Scientific Computation. 2004.

[39] D. L. Donoho, “De-noising by soft-thresholding,” IEEE Trans. Inf. Theory, vol. 41, no. 3, pp. 613–627, Mar. 1995.

[40] 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 Trans. Image Process., vol. 20, no. 11, pp. 3097–3111, Nov. 2011.

[41] A. Sobral, T. Bouwmans, and E.-H. Zahzah, “LRSLibrary: Low-rank and sparse tools for background modeling and subtraction in videos,” in Robust Low-Rank and Sparse Matrix Decomposition: Applications in Image and Video Processing, 2016.

[42] M. Tao, J. Yang, and B. He, “Alternating direction algorithms for total variation deconvolution in image reconstruction,” Dept. Math., Nanjing Univ., Nanjing, China, Tech. Rep. TR0918, 2009.

[43] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, “Image quality assessment: From error visibility to structural similarity,” IEEE Trans. Image Process., vol. 13, no. 4, pp. 600–612, Apr. 2004.

[44] V. R. Joseph and Y. Hung, “Orthogonal-maximin Latin hypercube designs,” Statistica Sinica, vol. 18, no. 1, pp. 171–186, 2008.

[45] L. Maddalena and A. Petrosino, “Towards benchmarking scene background initialization,” in Proc. Int. Conf. Image Anal. Process. Springer, 2015, pp. 469–476.

[46] 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.

Bo Shen (Graduate Student Member, IEEE) received the B.S. degree in statistics from the University of Science and Technology of China, Hefei, China, in July 2017. He is currently pursuing the Ph.D. degree in industrial and systems engineering with Virginia Tech, Blacksburg, VA, USA. His research interests include optimization in machine learning and data analytics in smart manufacturing.

Rakesh R. Kamath received the M.E. degree in materials engineering from the Indian Institute of Science, Bengaluru, India, in 2016. He is currently pursuing the Ph.D. degree in materials science and engineering with The University of Tennessee, Knoxville, USA. His current research interests involve 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, The 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 (Member, IEEE) received the B.S. and M.S. degrees in mechanical engineering from the Harbin Institute of Technology, Harbin, China, in 1993 and 1995, respectively, and the 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 modeling, synthesis, and diagnosis for large and complex manufacturing systems.