TR-ADMM: An inexact ADMM based robust tensor ring decomposition for video foreground detection

To Ngoc Hai*, Thanh Trung Le*, Viet-Dung Nguyen†, Nguyen Linh Trung*, Karim Abed-Meraim‡

* VNU University of Engineering and Technology, Vietnam National University, Hanoi, Vietnam

† ENSTA Bretagne, France

‡ University of Orléans, France

***Abstract*—Tensor ring (TR) decomposition provides an attractive tensor model for representing latent low-rank structures underlying in multivariate and high-dimensional data. Existing TR methods, however, often suffer from high computational complexity and/or sensitivity to data corruption. To address these limitations, we propose TR-ADMM, an effective robust tensor ring decomposition method that integrates an inexact alternating direction method of multipliers with a randomized sketching technique. Our method is evaluated on real-world video background and foreground separation tasks, where the video background is represented by low-rank TR components and moving objects are modeled as sparse outliers. Extensive experimental results demonstrate that TR-ADMM achieves competitive, in many cases, superior performance as compared with state-of-the-art tensor-based methods.**

***Index Terms*—Tensor decomposition, tensor ring, ADMM, randomized methods, video background and foreground.**

I. Introduction

Tensors are multiway arrays that provide a natural representation for multivariate and high-dimensional datasets [1]. Tensor decomposition (TD), which factorizes tensors into simpler components, has become a powerful analytical tool for uncovering latent structures in both batch and streaming data [2]–[4]. In practice, TD has been successfully applied in several applications, such as hyperspectral data processing [5], biomedical signal/image analysis [6], and video background-foreground separation [7], among others.

Compared to current TD models, tensor ring (TR) decomposition has emerged as an attractive representation framework for high-order tensors, offering several appealing properties [8]. Specifically, an nth-order tensor $\mathcal{X}$ (with $n \geq 3$) can be represented in the TR model by a set of n third-order core tensors $\{\mathcal{G}\}_{k=1}^{n}$ arranged in a circular topology. Under this format, the storage complexity scales as $\mathcal{O}(Inr^2)$ (r being the tensor rank), which is linear in both the tensor dimension I and order n. Consequently, TR provides one of the most memory-efficient representations for high-order tensors, ranking second only to the CP/PARAFAC decomposition [9]. Although CP achieves the lowest storage complexity of $\mathcal{O}(Inr)$, computing the CP decomposition of a given tensor can be NP-hard [10]. In contrast, its TR decomposition always exists. Moreover, unlike Tucker/HOSVD [11] and more recent tensor models such as fully connected tensor network [12] and tensor wheel [13], TR does not suffer from the curse of dimensionality, as the storage complexity of these formats grows exponentially with the tensor order. Compared with tensor train (TT) decomposition [14], which has the same storage complexity, TR offers greater representational flexibility due to its network topology. This format introduces inherent circular shift invariance and enables a circular unfolding scheme, which can be advantageous in practice. Consequently, TR has attracted increasing attention for higher-order and large-scale tensor decomposition problems.

In practice, data imperfections remain a fundamental challenge for tensor decomposition in general and TR models in particular. Most existing TR methods are based on alternating least squares (ALS), sequential singular value decompositions (SVD), or their variants, see [8], [15]–[17]. However, they often exhibit slow convergence and suboptimal estimation accuracy when applied to large-scale data that are noisy or heavily corrupted. To address these limitations, some robust TR methods have been proposed in the literature [18]–[22]. Specifically, the work in [18] introduced a convex ℓ_1-norm-regularized tensor ring nuclear norm method. Building on this framework, [19] extended the model to handle multi-mode tensor group sparsity via $\ell_{2,1}$-norm regularization. Instead of the ℓ_1-norm, the method in [20] employed a smooth and convex $\ell_{p,\epsilon}$-norm regularization to better mitigate data corruption. In [21], a correntropy-based criterion was proposed to reduce the influence of outliers. Meanwhile, [22] developed a Bayesian-based robust TR method. Despite their theoretical foundations, these methods often struggle to yield satisfactory performance in modern applications, such as video background modeling and foreground detection.

The main contributions of this work lie in the integration of three key technical enhancements which are designed to improve both the efficiency and robustness of TR decomposition. First, we develop an inexact ADMM framework utilizing a relaxed dual update strategy, which ensures stable convergence and high numerical accuracy. Secondly, to model the sparse corruption more effectively, our framework is coupled with a nonconvex ℓ_p-norm regularization ($0 < p < 1$), providing a more powerful sparsity-promoting mechanism than classical convex methods. Third, we integrate a randomized sketching mechanism into the TR core update step, significantly reducing the computational cost for high-dimensional tensor data without compromising decomposition accuracy. Experiments on real-world video background-foreground separation tasks

Corresponding author: Thanh Trung Le (thanhletrung@vnu.edu.vn)

TABLE I: Notational conventions

Notations	Descriptions
$x, \mathbf{x}, \mathbf{X}, \mathcal{X}$	scalar, vector, matrix, and tensor
$\mathcal{X}(i_1, i_2, \ldots, i_N)$	$(i_1, i_2, \ldots i_N)$-th element of $\mathcal{X}$
$\mathbf{X}^{-1}$, $\mathbf{X}^\top$, $\mathrm{tr}(\mathbf{X})$	inverse, transpose, and trace of $\mathbf{X}$
$\underline{\mathbf{X}}_{\langle k \rangle}$	mode-k unfolding of $\mathcal{X}$
$\mathrm{reshape}\{\mathcal{X}, [\mathbf{a}]\}$	reshape $\mathcal{X}$ into a tensor of size $a_1 \times \cdots \times a_N$
$\circ, \langle \cdot \rangle$	outer and inner product
$\|.\|_p$, $\|.\|_F$	ℓ_p norm and Frobenius norm
$\mathcal{X} = \mathfrak{TR}\big(\{\mathcal{G}_k\}_{k=1}^n\big)$	tensor ring decomposition of $\mathcal{X}$

demonstrate that the proposed method consistently outperforms existing state-of-the-art tensor approaches in terms of both separation quality and computational efficiency.

The commonly used mathematical notations and operators are summarized in Table I.

II. Problem Formulation

Given an nth-order tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_n}$, its tensor ring (TR) decomposition can be expressed via circular multilinear products over a set of third-order core tensors [8], as follows:

$$\begin{aligned}\mathcal{X} &= \sum_{\alpha_1, \ldots, \alpha_n = 1}^{r_1, \ldots, r_n} \mathcal{G}_1(\alpha_1, :, \alpha_2) \circ \mathcal{G}_2(\alpha_2, :, \alpha_3) \circ \cdots \circ \mathcal{G}_n(\alpha_n, :, \alpha_1) \\ &= \mathfrak{TR}\big(\{\mathcal{G}_k\}_{k=1}^n\big). \end{aligned} \tag{1}$$

Here, "$\circ$" denotes the outer product; $\mathcal{G}_k \in \mathbb{R}^{r_k \times I_k \times r_{k+1}}$ is the kth TR core, $\mathcal{G}_k(\alpha_k, :, \alpha_{k+1})$ denotes a vertical (mode-2) fiber; and the vector $\mathbf{r} = [r_1, r_2, \ldots, r_n]$ is called the *TR rank* (note that $r_{n+1} = r_1$). The mode-k unfolding matrix $\underline{\mathbf{X}}_{\langle k \rangle}$ of $\mathcal{X}$ is determined as

$$\underline{\mathbf{X}}_{\langle k \rangle} = \underline{\mathbf{G}}_{k,\langle 2 \rangle} \big(\underline{\mathbf{G}}_{\neq k,\langle 2 \rangle}\big)^\top, \tag{2}$$

where $\underline{\mathbf{G}}_{k,\langle 2 \rangle}$ and $\underline{\mathbf{G}}_{\neq k,\langle 2 \rangle}$ are, respectively, the mode-2 unfolding matrices of $\mathcal{G}_k$ and $\mathcal{G}_{\neq k} \in \mathbb{R}^{R_{k+1} \times \prod_{i=1, \neq k}^n I_i \times r_k}$ which is obtained by contracting all TR core tensors except $\mathcal{G}_k$. Thanks to the expression (2), the alternating least-squares (ALS) and its variants become the "workhorse" approach for estimating TR decomposition [8].

Assuming that the data tensor $\mathcal{X}$ is corrupted by a sparse component $\mathcal{S}$ and an additive Gaussian noise $\mathcal{N}$, the TR decomposition model (1) becomes

$$\mathcal{X} = \mathfrak{TR}\big(\{\mathcal{G}_k\}_{k=1}^n\big) + \mathcal{S} + \mathcal{N}. \tag{3}$$

To handle (3), robust decomposition methods are expected to promote sparsity, improve estimation accuracy and robustness against data corruption. In this work, we adopt the following optimization problem

$$\underset{\{\mathcal{G}_k\}_{k=1}^n, \mathcal{S}}{\arg\min} \frac{1}{2} \big\| \mathcal{X} - \mathfrak{TR}\big(\{\mathcal{G}_k\}_{k=1}^n\big) - \mathcal{S} \big\|_F^2 + \lambda \big\| \mathcal{S} \big\|_p^p, \tag{4}$$

where $\|\mathcal{S}\|_p^p = \sum_{i_1, i_2, \ldots, i_n} |\mathcal{S}(i_1, i_2, \ldots, i_n)|^p$, and λ is a regularization parameter to control the sparsity level. In particular, when $p = 1$, this term boils down to the widely used ℓ_1-norm regularization in robust tensor decomposition and completion (see, e.g., [18], [19], [23], [24]). When $p < 1$, the ℓ_p-norm promotes sparsity more aggressively and is then especially effective in capturing sharp and localized outliers, such as moving objects in video surveillance sequences. However, in such cases, (4) becomes nonconvex, which requires elegant optimization strategies. In the next section, we present an efficient approach for minimizing (4) effectively.

Algorithm 1: TR-ADMM: Robust TR decomposition via inexact-ADMM and randomized sketching

Input: Tensor $\mathcal{X}$, TR-ranks $\mathbf{r}$
Initialize: Generate $\{\mathcal{G}_k\}_{k=1}^n$ at random or zero; set $\mathcal{S} = 0, \mathcal{H} = 0$, and $\mathcal{Y} = 0$;
while *not converged* **do**
  for $k = 1$ **to** n **do**
    Update $\mathcal{G}_k$ according to (8);
  end
  Update $\mathcal{S}$ according to (12);
  Update $\hat{\mathcal{H}}$ according to (14);
  Update $\hat{\mathcal{Y}}$ according to (15);
  Compute check value δ according to (16);
  if $\delta \le 0$ **then**
    $\mathcal{H} = \hat{\mathcal{H}}$, $\mathcal{Y} = \hat{\mathcal{Y}}$;
  end
  else
    Update $\mathcal{H}$ and $\mathcal{Y}$ according to (17);
  end
end
return $\{\mathcal{G}_k\}_{k=1}^n$, $\mathcal{S}$

III. Proposed Algorithm

In this section, we propose an effective optimization framework based on an inexact alternating direction method of multipliers (inexact ADMM). Unlike standard ADMM, our method solves subproblems approximately in each iteration, reducing computational cost without sacrificing convergence. To estimate the sparse component $\mathcal{S}$, we introduce an auxiliary tensor $\mathcal{H}$ of the same size and a dual multiplier $\mathcal{Y}$ with a penalty parameter $\mu > 0$. Specifically, our objective function (4) is recast into the following ADMM form

$$\begin{aligned} &\underset{\{\mathcal{G}_k\}_{k=1}^n, \mathcal{S}, \mathcal{H}}{\mathrm{argmin}} \frac{1}{2} \big\| \mathcal{X} - \mathfrak{TR}\big(\{\mathcal{G}_k\}_{k=1}^n\big) - \mathcal{S} \big\|_F^2 + \lambda \big\| \mathcal{H} \big\|_p^p, \\ &\text{subject to} \quad \mathcal{S} - \mathcal{H} = 0. \end{aligned} \tag{5}$$

and its corresponding augmented Lagrangian is given by

$$\begin{aligned} L_\mu\big(\{\mathcal{G}_k\}_{k=1}^n, \mathcal{S}, \mathcal{H}, \mathcal{Y}\big) &= \frac{1}{2} \big\| \mathcal{X} - \mathfrak{TR}\big(\{\mathcal{G}_k\}_{k=1}^n\big) - \mathcal{S} \big\|_F^2 \\ &+ \lambda \|\mathcal{H}\|_p^p + \big\langle \mathcal{Y}, \mathcal{S} - \mathcal{H} \big\rangle + \frac{\mu}{2} \|\mathcal{S} - \mathcal{H}\|_F^2 . \end{aligned} \tag{6}$$

Here, our optimization framework offers an effective iterative procedure for finding a stationary point of (5) by decomposing it into four subproblems, namely: (i) updating the TR cores $\{\mathcal{G}_k\}_{k=1}^n$, (ii) updating the sparse component $\mathcal{S}$, (iii) updating the auxiliary variable $\mathcal{H}$, and (iv) updating the dual $\mathcal{Y}$. We refer to this method as TR-ADMM and summarize it in Algorithm 1. In what follows, we describe how TR-ADMM solves each subproblem in details.

A. Update of TR Cores $\{\mathcal{G}_k\}_{k=1}^n$

At the $(\ell + 1)$-th iteration, we can minimize the augmented Lagrangian function $L(\cdot)$ with respect to (w.r.t.) $\mathcal{G}_k$ while the

remaining cores and variables $\mathcal{S}$, $\mathcal{H}$, and $\mathcal{Y}$ are fixed. This results in the following ordinary least-squares (LS) problem

$$\mathbf{G}_k^{(\ell+1)} = \underset{\mathbf{G}_k}{\operatorname{argmin}} \frac{1}{2} \left\| \underline{\mathbf{X}}_{\langle k \rangle} - \underline{\mathbf{S}}_{\langle k \rangle}^{(\ell)} - \mathbf{G}_k \left(\underline{\mathbf{G}}_{\neq k,\langle 2 \rangle}^{(\ell)} \right)^\top \right\|_F^2, \quad (7)$$

$$\mathcal{G}_k^{(\ell+1)} = \operatorname{reshape} \left\{ \mathbf{G}_k^{(\ell+1)}, [r_k, I_k, r_{k+1}] \right\}, \quad (8)$$

Computing the exact LS solution of (7) requires the computation of the inverse of the associated Gram matrix. However, this step is computationally prohibitive, especially for high-dimensional tensor data. To overcome this issue, TR-ADMM employs the following randomized LS technique [25].

We exploit the fact that the tensor dimensions are generally much larger than the TR ranks, i.e., $\prod_{i=1,\neq k}^{n} I_i \gg r_k r_{k+1}$ for all k. Consequently, (7) constitutes an overdetermined least-squares problem and can be efficiently solved via matrix sketching. Specifically, by introducing a random Gaussian matrix $\mathbf{\Omega}$ of size $(\prod_{i=1,\neq k}^{n} I_i) \times M$ where $\prod_{i=1,\neq k}^{n} I_i \gg M \geq r_k r_{k+1}$, the k-th TR core $\mathcal{G}_k$ can be efficiently computed from

$$\mathbf{G}_k^{(\ell+1)} = \underset{\mathbf{G}_k}{\operatorname{argmin}} \frac{1}{2} \left\| \widetilde{\mathbf{X}}^{(\ell)} - \mathbf{G}_k \widetilde{\mathbf{Z}}^{(\ell)} \right\|_F^2, \quad (9)$$

where $\widetilde{\mathbf{X}}^{(\ell)} = \left(\underline{\mathbf{X}}_{\langle k \rangle} - \underline{\mathbf{S}}_{\langle k \rangle}^{(\ell)} \right) \mathbf{\Omega}$ and $\widetilde{\mathbf{Z}}^{(\ell)} = \left(\underline{\mathbf{G}}_{\neq k,\langle 2 \rangle}^{(\ell)} \right)^\top \mathbf{\Omega}$. In particular, the solution of (9) can be determined as

$$\mathbf{G}_k^{(\ell+1)} = \left(\widetilde{\mathbf{X}}^{(\ell)} \left(\widetilde{\mathbf{Z}}^{(\ell)} \right)^\top \right) \left(\widetilde{\mathbf{Z}}^{(\ell)} \left(\widetilde{\mathbf{Z}}^{(\ell)} \right)^\top + \epsilon \mathbf{I} \right)^{-1}, \quad (10)$$

where $\epsilon > 0$ is a small regularization parameter and $\mathbf{I}$ is the identity matrix. This regularization term can ensure that the matrix inversion remains numerically stable and effectively avoids singularity issues during the optimization process.

B. Update of the sparse component $\mathcal{S}$

The sparse component $\mathcal{S}^{(\ell+1)}$ is updated by minimizing $L(\cdot)$ in (6) w.r.t. $\mathcal{S}$ as follows:

$$\mathcal{S}^{(\ell+1)} = \underset{\mathcal{S}}{\operatorname{argmin}} \frac{1}{2} \left\| \mathcal{X} - \mathcal{L}^{(\ell+1)} - \mathcal{S} \right\|_F^2 + \langle \mathcal{Y}^{(\ell)}, \mathcal{S} - \mathcal{H}^{(\ell)} \rangle + \frac{\mu}{2} \| \mathcal{S} - \mathcal{H}^{(\ell)} \|_F^2, \quad (11)$$

where $\mathcal{L}^{(\ell+1)} = \mathfrak{TR}\left(\{\mathcal{G}_k^{(\ell+1)}\}_{k=1}^n \right)$. As all terms of (11) are convex, its closed-form solution is then given by

$$\mathcal{S}^{(\ell+1)} = \frac{1}{1+\mu} \left(\mathcal{X} - \mathcal{L}^{(\ell+1)} + \mu \left(\mathcal{H}^{(\ell)} - \mathcal{Y}^{(\ell)} \right) \right). \quad (12)$$

C. Update of the auxiliary variable $\mathcal{H}$

Similar to the update of $\mathcal{S}^{(\ell+1)}$, $\mathcal{H}^{(\ell+1)}$ is obtained from

$$\mathcal{H}^{(\ell+1)} = \underset{\mathcal{H}}{\operatorname{argmin}} \frac{1}{2} \left\| \mathcal{H} - \left(\mathcal{S}^{(\ell+1)} + \frac{\mathcal{Y}^{(\ell)}}{\mu} \right) \right\|_F^2 + \frac{\lambda}{\mu} \| \mathcal{H} \|_p^p. \quad (13)$$

Here, when $0 < p < 1$, (13) becomes nonconvex. To efficiently solve this problem, we employ the Generalized Soft-Thresholding (GST) operator [26] to obtain the solution for $\mathcal{H}$, as defined below

$$\hat{\mathcal{H}}^{(\ell+1)} = \mathrm{GST}_p \left(\mathcal{S}^{(\ell+1)} + \frac{\mathcal{Y}^{(\ell)}}{\mu}, \frac{\lambda}{\mu}, J \right), \quad (14)$$

where J denotes the number of GST's iterations,[1] see Algorithm 2 for an element-wise version. The solution $\hat{\mathcal{H}}^{(\ell+1)}$ in (14) represents an intermediate estimate of the auxiliary tensor before it is incorporated into the inexact ADMM update in the next stage.

[1] As suggested in [26], reasonable performance can be obtained by choosing a small number of iterations, typically $J = 2$ or 3 in practice.

Algorithm 2: GST: Generalized Soft Thresholding

Input: z, λ, p, J

$\tau_p^{\mathrm{GST}}(\lambda) = \left(2\lambda(1-p)\right)^{\frac{1}{2-p}} + \lambda p \left(2\lambda(1-p)\right)^{\frac{p-1}{2-p}}$

if $|z| \leq \tau_p^{\mathrm{GST}}(\lambda)$ **then**

  $\hat{z} = 0$

end

else

  $x^{(0)} = |z|$

  for $k = 0$ **to** J **do**

    $x^{(k+1)} = |z| - \lambda p (x^{(k)})^{p-1}$

  end

  $\hat{z} = \operatorname{sign}(z) x^{(J)}$

end

return $\hat{z} = \mathrm{GST}_p(z, \lambda, J)$;

D. Update of the dual $\mathcal{Y}$

In this stage, an inexact strategy is implemented to enhance the efficiency of TR-ADMM. First, the intermediate dual variable $\hat{\mathcal{Y}}^{(\ell+1)}$ is updated as follows

$$\hat{\mathcal{Y}}^{(\ell+1)} = \mathcal{Y}^{(\ell)} + \mu (\mathcal{S}^{(\ell+1)} - \hat{\mathcal{H}}^{(\ell+1)}). \quad (15)$$

Next, we define a criterion to measure the estimation accuracy:

$$\delta = \left\langle \mu (\mathcal{Y}^{(\ell)} - \hat{\mathcal{Y}}^{(\ell+1)}), -(\mathcal{H}^{(\ell)} - \hat{\mathcal{H}}^{(\ell+1)}) \right\rangle. \quad (16)$$

If $\delta \leq 0$, the intermediate estimates are accepted and we set $\mathcal{H}^{(\ell+1)} = \hat{\mathcal{H}}^{(\ell+1)}$ and $\mathcal{Y}^{(\ell+1)} = \hat{\mathcal{Y}}^{(\ell+1)}$. Otherwise, a refinement step is performed:

$$\mathcal{H}^{(\ell+1)} = \mathcal{H}^{(\ell)} - \gamma (\mathcal{H}^{(\ell)} - \hat{\mathcal{H}}^{(\ell+1)}), \quad (17)$$

$$\mathcal{Y}^{(\ell+1)} = \mathcal{Y}^{(\ell)} - \gamma (\mathcal{Y}^{(\ell)} - \hat{\mathcal{Y}}^{(\ell+1)}), \quad (18)$$

where $\gamma > 1$ is a constant scaling factor.

E. Stopping Conditions and Computational Complexity

Our procedure stops when $\max(\|\mathcal{X} - \mathfrak{TR}(\{\mathcal{G}_k\}_{k=1}^n) - \mathcal{S}\|_F / \|\mathcal{X}\|_F, \|\mathcal{S} - \mathcal{H}\|_F / \|\mathcal{S}\|_F) < \sigma$ or the maximum number of iterations is reached, where σ is a predefined tolerance.

For simplicity, we assume that TR ranks $r_1 = \cdots = r_n = r$ and tensor dimensions $I_1 = \cdots = I_n = I$. Updating the TR cores $\{\mathcal{G}^{(k)}\}$ requires a total complexity of $\mathcal{O}(nr^4 I^{n-1} M)$ flops, where $M = sr^2$ and s denotes the sketch rate used in the randomized least-squares steps. The updates of the sparse component $\mathcal{S}$ and the dual variable $\mathcal{Y}$ consist of element-wise operations and thus incur a cost of $\mathcal{O}(I^n)$ flops. Updating the auxiliary variable $\mathcal{H}$ requires $\mathcal{O}(I^n J)$ flops, where J is a small number of GST iterations. In practice, we typically set $s = 5$ and $J = 2$ or 3. Consequently, the overall computational complexity of TR-ADMM is $\mathcal{O}\left(I^{n-1} \max(I, nr^4 M)\right)$ flops.

IV. Numerical Experiments

In this section, we demonstrate the effectiveness of the proposed method for the task of video foreground detection, in comparison with the state-of-the-art tensor-based methods. They include RTCUR [27], HQ-TCAD [28], RC-FCTN [29], RTRC [18], and STRC [7].

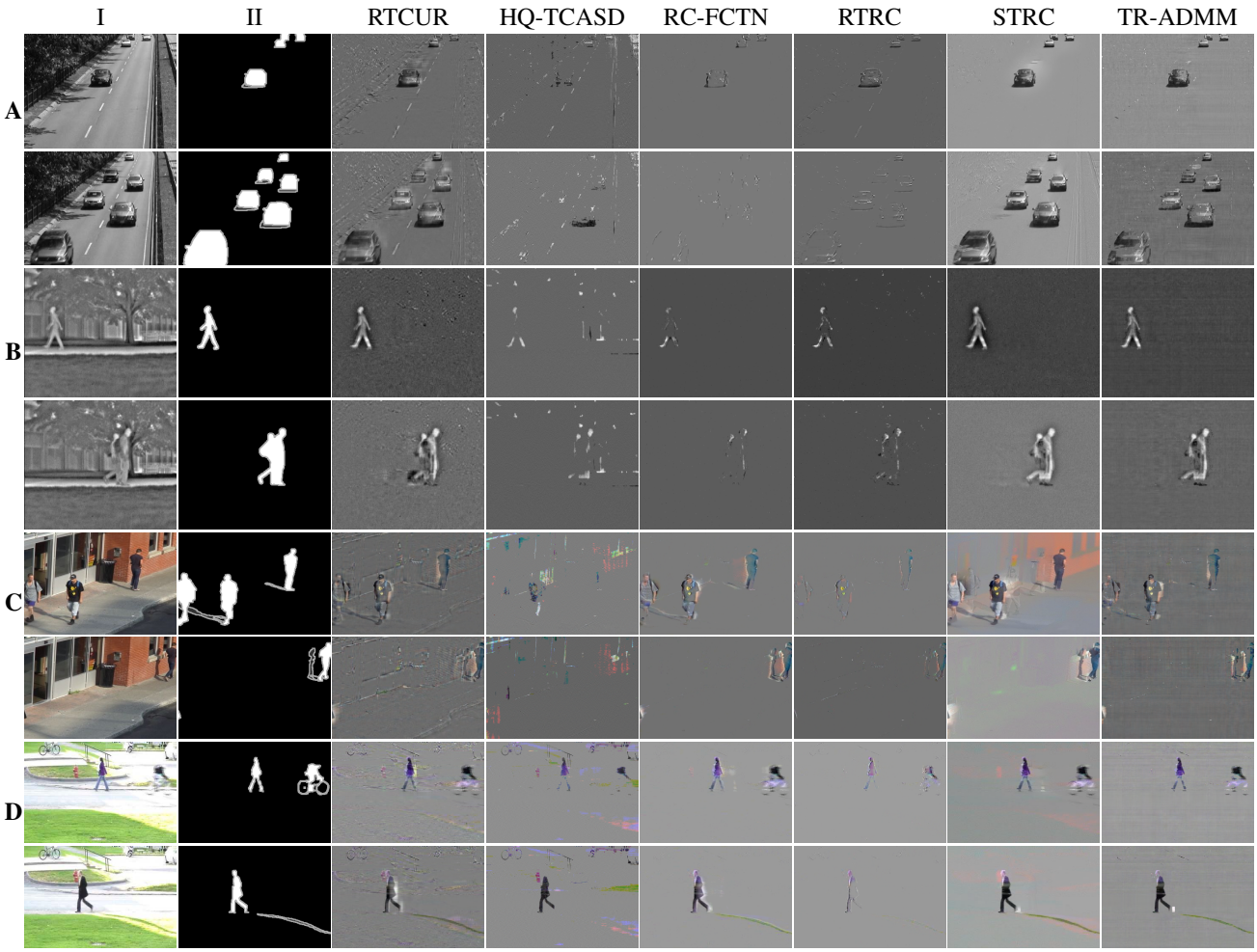

Fig. 1: Video Foreground Detection Results. Columns I and II represent the original data and ground truth. Datasets: (A) `Highway`, (B) `Park`, (C) `busStation` and (D) `Pedestrians`.

TABLE II: Performance comparison on different video datasets. PRE: Precision, REC: Recall, F1: F1-score.

Datasets	**Highway**			**Park**			**busStation**			**Pedestrians**		
Metrics	**PRE**	**REC**	**F1**	**PRE**	**REC**	**F1**	**PRE**	**REC**	**F1**	**PRE**	**REC**	**F1**
RTCUR	0.13	0.80	0.22	0.29	0.74	0.42	0.27	0.79	0.40	0.08	0.97	0.15
HQ-TCASD	0.11	0.13	0.12	0.27	0.31	0.29	0.18	0.13	0.15	0.14	0.92	0.24
RC-FCTN	**0.91**	0.26	0.40	**0.95**	0.06	0.12	0.54	0.50	0.52	**0.94**	0.53	0.68
RTRC	0.62	0.40	0.46	0.80	0.19	0.30	**0.71**	0.14	0.23	0.84	0.35	0.50
SRTC	0.42	**0.82**	0.56	0.44	**0.75**	0.56	0.19	**0.97**	0.31	0.16	**0.97**	0.28
TR-ADMM	0.74	0.62	**0.68**	0.94	0.69	**0.79**	0.63	0.81	**0.71**	0.65	0.95	**0.77**

Datasets. Our experiments are conducted on four real video sequences from the CDnet2014 dataset [30]. They include a baseline video (`Highway`), a thermal video (`Park`), and two color videos with shadows and shades (`busStation` and `Pedestrians`). The video resolutions of these sequences are 240×320, 288×352, and 360×240, and 240×360, respectively. To reduce the computational burden of tensor-based methods while still achieving a representative evaluation performance, we use the first 50 frames of each sequence to construct the corresponding video tensor.

Parameter Settings. The hyperparameters of TR-ADMM are carefully fine-tuned via cross-validation on a validation subset and then fixed for all testing scenarios. In particular, for all benchmark video sequences, the TR rank parameters are set to 10×10 for the horizontal and vertical dimensions, 5 for the spatial dimension, and 3 for the color channel.

In the randomized sketching step, the sketch rate s is fixed to 5. The sparsity regularization parameter is chosen as $\lambda = 1/\sqrt{\max(\{I_i\}_{i=1}^n)}$ to balance the low-rank structure preservation and sparse outlier detection. For the ℓ_p-norm regularization, we set $p = 0.5$ to promote the sparsity in the video foreground components (i.e. moving objects). This fixed parameter configuration consistently results in reasonable performance across all three sequences. For all other competing tensor-based methods, the default parameter settings reported in the original papers are used.

Evaluation Metrics. To measure the detection performance of algorithms, we use three metrics: Precision, Recall, and F1-score. These metrics are computed based on four pixel-wise classifications: true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN), as follows:

$$\text{Precision} = \frac{\text{TP}}{\text{TP}+\text{FP}}, \quad \text{Recall} = \frac{\text{TP}}{\text{TP}+\text{FN}}, \tag{19}$$

$$\text{F1-score} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}. \tag{20}$$

Here, Precision evaluates the accuracy of foreground separation, while Recall measures the ability to correctly recover true foreground regions. The F1-score, defined as the harmonic mean of Precision and Recall, provides a balanced performance indicator. Higher values of these metrics indicate better algorithm performance.

Experimental Results. Fig. 1 illustrates the visual foreground detection results of the algorithms, while Table II reports their performance in terms of Precision, Recall, and F1-score. We can see that TR-ADMM outperforms other algorithms, achieving the highest F1-score in all four datasets with 0.68 for `Highway`, 0.79 for `Park`, 0.71 for `busStation`, and 0.77 for `Pedestrians`. Although STRC attains the highest Recall on all datasets, it has a relatively low Precision (i.e., ≤ 0.6 in all testing). The remaining algorithms also exhibit a significant gap between their Precision and Recall values. For example, RC-FCTN reaches the highest Precision of 0.91 but only 0.26 Recall on `Highway`; similarly, it achieves a peak Precision of 0.95 on `Park` with a minimum Recall of 0.06. In contrast, our method maintains stable high performance in all three evaluation metrics. It indicates that TR-ADMM consistently outperforms other methods, demonstrating the effectiveness of our optimization strategies.

V. Conclusions

In this work, we have proposed a new efficient framework for background and foreground separation in real-world video sequences based on robust tensor ring decomposition. Experimental results demonstrate that our approach consistently outperforms several state-of-the-art tensor-based methods. Future research will focus on extending this framework to handle real-time video streaming and exploring more advanced tensor network topologies to further enhance performance.

References

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

[2] G. Ballard and T. G. Kolda, *Tensor Decompositions for Data Science*. Cambridge University Press, 2025.

[3] Y. Liu, J. Liu, Z. Long, and C. Zhu, *Tensor Computation for Data Analysis*. Springer, 2022.

[4] T. T. Le *et al.*, "A contemporary and comprehensive survey on streaming tensor decomposition," *IEEE Trans. Knowl. Data Eng.*, vol. 35, no. 11, pp. 10 897–10 921, 2023.

[5] Y.-J. Deng *et al.*, "Tensor decomposition-based relaxed linear regression for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 63, pp. 1–16, 2025.

[6] T. T. Le *et al.*, "Tensor-based higher-order multivariate singular spectrum analysis and applications to multichannel biomedical signal analysis," *Signal Process.*, vol. 238, p. 110113, 2026.

[7] B. Shen, W. Xie, and Z. J. Kong, "Smooth robust tensor completion for background/foreground separation with missing pixels: Novel algorithm with convergence guarantee," *J. Mach. Learn. Res.*, vol. 23, no. 217, pp. 1–40, 2022.

[8] Q. Zhao, G. Zhou, S. Xie, L. Zhang, and A. Cichocki, "Tensor ring decomposition," *arXiv preprint arXiv:1606.05535*, 2016.

[9] R. Bro, "PARAFAC. Tutorial and applications," *Chemometr. Intell. Lab. Syst.*, vol. 38, no. 2, pp. 149–172, 1997.

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

[11] L. De Lathauwer, B. De Moor, and J. Vandewalle, "On the best rank-1 and rank-(r_1,r_2,...,r_n) approximation of higher-order tensors," *SIAM J. Matrix Anal. Appl.*, vol. 21, pp. 1324–1342, 2000.

[12] Y.-B. Zheng *et al.*, "Fully-connected tensor network decomposition and its application to higher-order tensor completion," in *Proc. AAAI*, vol. 35, no. 12, 2021, pp. 11 071–11 078.

[13] Z.-C. Wu *et al.*, "Tensor wheel decomposition and its tensor completion application," in *Proc. NeurIPS*, vol. 35, 2022, pp. 27 008–27 020.

[14] I. V. Oseledets, "Tensor-train decomposition," *SIAM J. Sci. Comput.*, vol. 33, no. 5, pp. 2295–2317, 2011.

[15] W. Wang, V. Aggarwal, and S. Aeron, "Efficient low rank tensor ring completion," in *Proc. ICCV*, 2017, pp. 5697–5705.

[16] L. Yuan, C. Li, J. Cao, and Q. Zhao, "Randomized tensor ring decomposition and its application to large-scale data reconstruction," in *Proc. IEEE ICASSP*, 2019, pp. 2127–2131.

[17] O. Mickelin and S. Karaman, "On algorithms for and computing with the tensor ring decomposition," *Numer. Linear Alg. Appl.*, vol. 27, no. 3, p. e2289, 2020.

[18] H. Huang *et al.*, "Robust low-rank tensor ring completion," *IEEE Trans. Comput. Imaging*, vol. 6, pp. 1117–1126, 2020.

[19] Y. Qiu *et al.*, "Towards multi-mode outlier robust tensor ring decomposition," in *Proc. AAAI*, vol. 38, no. 13, 2024, pp. 14 713–14 721.

[20] X. P. Li and H. C. So, "Robust low-rank tensor completion based on tensor ring rank via $\ell_{p,\epsilon}$-norm," *IEEE Trans. Signal Process.*, vol. 69, pp. 3685–3698, 2021.

[21] Y. He and G. K. Atia, "Scalable and robust tensor ring decomposition for large-scale data," in *Proc. UAI*, 2023, pp. 860–869.

[22] Z. Huang, Y. Qiu, X. Chen, W. Sun, and G. Zhou, "Bayesian robust tensor ring decomposition for incomplete multiway data," *IEEE Trans. Syst. Man Cybern.*, vol. 54, no. 7, pp. 4005–4018, 2024.

[23] G. Song, M. K. Ng, and X. Zhang, "Robust tensor completion using transformed tensor singular value decomposition," *Numer. Linear Algebra Appl.*, vol. 27, no. 3, p. e2299, 2020.

[24] T. T. Le *et al.*, "Robust tensor tracking with missing data and outliers: Novel adaptive CP decomposition and convergence analysis," *IEEE Trans. Signal Process.*, vol. 70, pp. 4305–4320, 2022.

[25] M. W. Mahoney, "Randomized algorithms for matrices and data," *Found. Trends Mach. Learn.*, vol. 3, no. 2, pp. 123–224, 2011.

[26] W. Zuo *et al.*, "A generalized iterated shrinkage algorithm for non-convex sparse coding," in *Proc. ICCV*, 2013, pp. 217–224.

[27] H. Cai, Z. Chao, L. Huang, and D. Needell, "Robust tensor cur decompositions: Rapid low-tucker-rank tensor recovery with sparse corruptions," *SIAM J. Imaging Sci.*, vol. 17, no. 1, pp. 225–247, 2024.

[28] Y. He and G. K. Atia, "Robust low-tubal-rank tensor completion based on tensor factorization and maximum correntopy criterion," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 35, no. 10, pp. 14 603–14 617, 2023.

[29] Y.-Y. Liu, X.-L. Zhao, G.-J. Song, Y.-B. Zheng, and T.-Z. Huang, "Fully-connected tensor network decomposition for robust tensor completion problem," *Inverse Probl. Imaging*, vol. 18, no. 1, pp. 208–238, 2024.

[30] Y. Wang *et al.*, "CDnet 2014: An expanded change detection benchmark dataset," in *Proc. CVPRW*, 2014, pp. 387–394.