

# Fast Robust Tensor Principal Component Analysis via Fiber CUR Decomposition \*

HanQin Cai    Zehan Chao    Longxiu Huang    Deanna Needell  
 Department of Mathematics  
 University of California, Los Angeles  
 Los Angeles, CA 90095, USA  
 {hqcai, zchao, huangl3, deanna}@math.ucla.edu

## Abstract

We study the problem of tensor robust principal component analysis (TRPCA), which aims to separate an underlying low-multilinear-rank tensor and a sparse outlier tensor from their sum. In this work, we propose a fast non-convex algorithm, coined Robust Tensor CUR (RTCUR), for large-scale TRPCA problems. RTCUR considers a framework of alternating projections and utilizes the recently developed tensor Fiber CUR decomposition to dramatically lower the computational complexity. The performance advantage of RTCUR is empirically verified against the state-of-the-arts on the synthetic datasets and is further demonstrated on the real-world application such as color video background subtraction.

## 1. Introduction

Robust principal component analysis (RPCA) [1] is one of the fundamental dimension reduction methods for data science. It tolerances extreme outliers, to which standard PCA is very sensitive. In particular, RPCA aims to reconstruct a low-rank matrix  $L_*$  and a sparse outlier matrix  $S_*$  from the corrupted observation

$$X = L_* + S_*. \quad (1)$$

This RPCA model has been widely studied [2–6] and applied to many applications e.g., face modeling [7], feature identification [8], NMR signal recovery [9], and video background subtraction [10]. However, RPCA can only handle 2-mode arrays (i.e., matrices) while real data usually presents in the form of a multi-dimensional array (i.e., tensors). For instance, in the application of video background subtraction [10], a color video is naturally a 4-mode tensor (height, width, frame, and color). To apply RPCA, the original tensor data has to be unfolded to a matrix along certain

mode(s), resulting in performance degradation due to the structural information loss at matricization. In addition, it may not be clear how the rank of the unfolded matrix depends on the rank of the original tensor in some tensor rank settings.

To take advantage of the natural tensor structure, some earlier studies have the extended RPCA model (1) to the so-called Robust Tensor PCA (RTPCA) model:

$$\mathcal{X} = \mathcal{L}_* + \mathcal{S}_*, \quad (2)$$

where  $\mathcal{L}_* \in \mathbb{R}^{d_1 \times \dots \times d_n}$  is the underlying low-rank tensor and  $\mathcal{S}_* \in \mathbb{R}^{d_1 \times \dots \times d_n}$  is the underlying sparse tensor. In this model, there is no assumption on the outliers' magnitudes; they can be arbitrary large as long as  $\mathcal{S}_*$  is sparse. Note that there exist various definitions of tensor decompositions that lead to various versions of tensor rank. In this work, we will focus on the setting using multilinear rank. Specifically speaking, we aim to solve the non-convex optimization problem:

$$\begin{aligned} & \underset{\mathcal{L}, \mathcal{S}}{\text{minimize}} \quad \|\mathcal{X} - \mathcal{L} - \mathcal{S}\|_F \\ & \text{subject to} \quad \mathcal{L} \text{ is low-multilinear-rank,} \\ & \quad \quad \quad \mathcal{S} \text{ is sparse.} \end{aligned} \quad (3)$$

One of the major challenges for solving the multilinear rank based TRPCA problem is the high computational cost for computing the Tucker decomposition. If  $\mathcal{L}_*$  is rank- $(r_1, \dots, r_n)$ , the existing methods, e.g., [11–14], have computational complexity at least  $\mathcal{O}(nd^n r)$  flops<sup>1</sup>—they are thus computationally challenging in large-scale<sup>2</sup> problems. Thus, it is urgent to develop a highly efficient TRPCA algorithm for time-intensive applications.

### 1.1. Related Work and Contributions

Due to the various definitions of tensor rank, there are different versions of robust tensor decompositions. For ex-

<sup>1</sup>For ease of presentation, we take  $d_1 = \dots = d_n =: d$  and  $r_1 = \dots = r_n =: r$  when discussing complexities throughout the paper.

<sup>2</sup>In our context, 'large-scale' refers to large  $n$  and/or large  $d$ .

\*This work is supported by NSF BIGDATA DMS #1740325 and NSF DMS #2011140.

ample, [15] proposes the tensor tubal rank based on the tensor SVD (t-SVD). Based on the tubal rank, the TRPCA is formulated by different convex optimization models with different sparsity patterns [16, 17]. This type of RTPCA only partially utilizes the inherent correlation of tensor data. Recently, based on the CP rank [18], [19] proposed a non-convex iterative algorithm to compute a robust tensor CP decomposition by assuming that the underlying tensor  $\mathcal{L}_\star$  is a low CP rank tensor and  $\mathcal{S}_\star$  is a sparse tensor.

In tensor analysis, another popular used tensor rank is multilinear rank [20]. Based on the multilinear rank, there is another version of RTPCA that considers the underlying low rank tensor  $\mathcal{L}_\star$  to be a low multilinear rank tensor. Motivated from the fact that the nuclear norm is the convex envelope of the matrix rank with the unit ball of the spectral norm, the work [21] proposed the Higher-order RPCA via replacing the multilinear rank of the tensor  $\mathcal{L}$  by the convex surrogate  $\sum_{i=1}^n \|\mathcal{L}_{(i)}\|_*$  and thus the convex optimization problem becomes

$$\begin{aligned} & \underset{\mathcal{L}, \mathcal{S}}{\text{minimize}} \quad \sum_{i=1}^n \|\mathcal{L}_{(i)}\|_* + \lambda \|\mathcal{S}\|_1 \\ & \text{subject to} \quad \mathcal{L} + \mathcal{S} = \mathcal{X}, \end{aligned} \quad (4)$$

where  $\|\cdot\|_*$  is matrix nuclear norm and  $\|\cdot\|_1$  is the entry-wise  $\ell_1$  norm.

In this work, we consider the TRPCA problem under the setting of multilinear rank. Our main contributions are two-fold:

1. We propose a novel non-convex approach, coined Robust Tensor CUR (RTCUR), for large-scale TRPCA problems. RTCUR uses an alternating projection framework and employs a novel mode-wise tensor decomposition [22] for fast low-rank tensor approximation. The computational complexity of RTCUR is as low as  $\mathcal{O}(n^2 d r^2 \log^2(d) + n^2 r^{n+1} \log^{n+1}(d))$ , which is substantially lower than the state-of-the-art.
2. The empirical advantages of RTCUR are verified on both synthetic and real-world datasets. In particular, we show that RTCUR has the best speed performance compared to the state-of-the-art of both tensor and matrix RPCA. Moreover, we also show that tensor methods outperform matrix methods, in terms of reconstruction quality, under certain outlier patterns.

## 1.2. Notation

We denote tensors, matrices, vectors, and scalars in different typeface for clarity. More specifically, calligraphic capital letters (e.g.,  $\mathcal{X}$ ) are used for tensors, capital letters (e.g.,  $X$ ) are used for matrices, lower boldface letters (e.g.,  $\mathbf{x}$ ) for vectors, and regular letters (e.g.,  $x$ ) for scalars. We use  $X(I, :)$  and  $X(:, J)$  to denote the row

and column submatrices with indices  $I$  and  $J$ , respectively.  $\mathcal{X}(I_1, \dots, I_n)$  denotes the subtensor of  $\mathcal{X}$  with indices  $I_k$  at mode  $k$ . A single element of a tensor is denoted by  $\mathcal{X}_{i_1, \dots, i_n}$ . Moreover,  $\|\mathcal{X}\|_\infty = \max_{i_1, \dots, i_n} |\mathcal{X}_{i_1, \dots, i_n}|$  and  $\|\mathcal{X}\|_F = \sqrt{\sum_{i_1, \dots, i_n} \mathcal{X}_{i_1, \dots, i_n}^2}$  denote the infinity norm and Frobenius norm, respectively.  $X^\dagger$  denotes the Moore-Penrose pseudoinverse. The set of the first  $d$  natural numbers is denoted by  $[d] := \{1, \dots, d\}$ .

## 2. Preliminaries

A tensor is a multidimensional array whose dimension is called the *order* or *mode*. The space of real tensors of order  $n$  and sizes  $(d_1, \dots, d_n)$  is denoted as  $\mathbb{R}^{d_1 \times \dots \times d_n}$ . We first review some basic tensor properties.

### 2.1. Tensor Operations

**Definition 2.1** (Tensor Matricization/Unfolding). *An  $n$ -mode tensor  $\mathcal{X}$  can be matricized, or reshaped into a matrix, in  $n$  ways by unfolding it along each of the  $n$  modes. The mode- $k$  matricization/unfolding of tensor  $\mathcal{X} \in \mathbb{R}^{d_1 \times \dots \times d_n}$  is the matrix denoted by*

$$\mathcal{X}_{(k)} \in \mathbb{R}^{d_k \times \prod_{j \neq k} d_j} \quad (5)$$

whose columns are composed of all the vectors obtained from  $\mathcal{X}$  by fixing all indices except for the  $k$ -th dimension. The mapping  $\mathcal{X} \mapsto \mathcal{X}_{(k)}$  is called the mode- $k$  unfolding operator.

**Definition 2.2** (Mode- $k$  Product). *Let  $\mathcal{X} \in \mathbb{R}^{d_1 \times \dots \times d_n}$  and  $A \in \mathbb{R}^{J \times d_k}$ . The  $k$ -th mode multiplication between  $\mathcal{X}$  and  $A$  is denoted by  $\mathcal{Y} = \mathcal{X} \times_k A$ , with*

$$\begin{aligned} & \mathcal{Y}_{i_1, \dots, i_{k-1}, j, i_{k+1}, \dots, i_n} \\ &= \sum_{s=1}^{d_k} \mathcal{X}_{i_1, \dots, i_{k-1}, s, i_{k+1}, \dots, i_n} A_{j, s}. \end{aligned} \quad (6)$$

Note this can be written as a matrix product by noting that  $\mathcal{Y}_{(k)} = A \mathcal{X}_{(k)}$ . If we have multiple tensor matrix products from different modes, we use the notation  $\mathcal{X} \times_{i=t}^s A_i$  to denote the product  $\mathcal{X} \times_t A_t \times_{t+1} \dots \times_s A_s$ .

**Definition 2.3** (Tucker decomposition and Tucker/Multilinear Rank). *Given an  $n$ -order tensor  $\mathcal{X}$ , the tuple  $(r_1, \dots, r_n) \in \mathbb{N}^n$  is called the Tucker rank of the tensor  $\mathcal{X}$ , where  $r_k = \text{rank}(\mathcal{X}_{(k)})$  i.e.,  $r_k$  is the column rank of the unfolding of the tensor  $\mathcal{X}$  from mode- $k$ . Its Tucker decomposition is defined as an approximation of a core tensor  $\mathcal{C} \in \mathbb{R}^{R_1 \times \dots \times R_n}$  multiplied by  $n$  factor matrices  $A_k \in \mathbb{R}^{d_k \times R_k}$  (whose columns are usually orthonormal),  $k = 1, 2, \dots, n$  along each mode, such that*

$$\mathcal{X} \approx \mathcal{C} \times_{i=1}^n A_i. \quad (7)$$

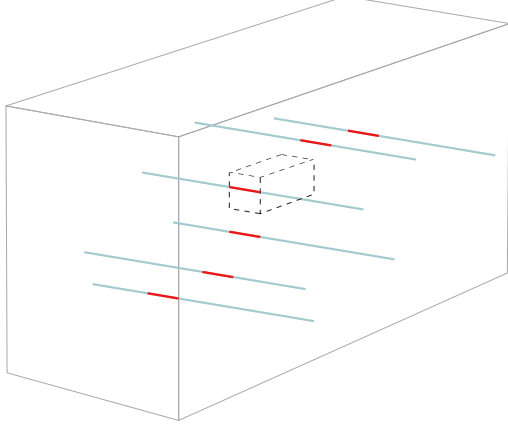


Figure 1: ([22, Figure 2]). Illustration of the Fiber CUR Decomposition of Theorem 2.5 in which  $J_i$  is not necessarily related to  $I_i$ . The lines correspond to rows of  $C_2$ , and red indices within correspond to rows of  $U_2$ . Note that the lines may (but do not have to) pass through the core subtensor  $\mathcal{R}$  outlined by dotted lines. Fibers used to form  $C_1$  and  $C_3$  are not shown for clarity.

If Equation (7) holds and  $R_k = r_k$  for all  $k = 1, \dots, n$ , then we call the decomposition an exact Tucker decomposition of  $\mathcal{X}$ .

**Remark 2.4.** Higher-order singular value decomposition (HOSVD) [23] is a specific orthogonal Tucker decomposition which is popularly used in the literature.

## 2.2. Tensor Fiber CUR Decompositions

CUR decompositions for matrices have been actively studied [24, 25]. For a matrix  $X \in \mathbb{R}^{d_1 \times d_2}$ , let  $C$  be a column submatrix of  $X$  with column indices  $J$ ,  $R$  be a row submatrix of  $X$  with row indices  $I$ , and  $U = X(I, J)$ . The theory of CUR decompositions states that  $X = CU^T R$  if  $\text{rank}(U) = \text{rank}(X)$ . The first extension of CUR decompositions to tensors involved a single-mode unfolding of 3-mode tensors [26]. Later, [27] proposes a different variant of tensor CUR that accounts for all modes. Recently, [22] dubs these decompositions with more descriptive monikers, namely Fiber and Chidori CUR decompositions. Later in this paper, we will employ the tensor Fiber CUR decomposition (see Figure 1 for illustration) to accelerate a key step in the proposed algorithm. For the reader's convenience, we state the characterization of the Fiber CUR decomposition below.

**Theorem 2.5** ([22, Theorem 3.3]). Let  $\mathcal{A} \in \mathbb{R}^{d_1 \times \dots \times d_n}$  with multilinear rank  $(r_1, \dots, r_n)$ . Let  $I_i \subseteq [d_i]$  and  $J_i \subseteq [\prod_{j \neq i} d_j]$ . Set  $\mathcal{R} = \mathcal{A}(I_1, \dots, I_n)$ ,  $C_i = \mathcal{A}_{(i)}(:, J_i)$  and  $U_i = C_i(I_i, :)$ . Then the following statements are equivalent:

- (i)  $\text{rank}(U_i) = r_i$ ,
- (ii)  $\mathcal{A} = \mathcal{R} \times_{i=1}^n (C_i U_i^\dagger)$ ,
- (iii)  $\text{rank}(C_i) = r_i$  for all  $i$  and the multilinear rank of  $\mathcal{R}$  is  $(r_1, \dots, r_n)$ .

In particular, (ii) is called tensor Fiber CUR decomposition.

In addition, according to [28, Corollary 5.2], if one uniformly samples indices  $I_i$  and  $J_i$  with size  $|I_i| = \mathcal{O}(r_i \log(d_i))$  and  $|J_i| = \mathcal{O}(r_i \log(\prod_{j \neq i} d_j))$ , then  $\text{rank}(U_i) = r_i$  holds for all  $i$  with high probability under some mild assumptions. Thus, the tensor Fiber CUR decomposition holds and its computational complexity is dominated by computing the pseudoinverse of  $U_i$ . Given the dimension of  $U_i$ , computing the pseudoinverse costs  $\mathcal{O}(r^2(n-1)\log^2(d))$  flops, thus tensor Fiber CUR decomposition costs  $\mathcal{O}(r^2 n^2 \log^2(d))$  flops. In contrast, HOSVD costs  $\mathcal{O}(rd^n)$  flops.

## 3. Proposed Approach

In this section, we propose a fast approach, called Robust Tensor CUR (RTCUR), for non-convex TRPCA problem (3). RTCUR is developed in a framework of alternating projections: (I) First, we project  $\mathcal{X} - \mathcal{L}^{(k)}$  onto the space of sparse tensors to update the estimate of outliers (i.e.,  $\mathcal{S}^{(k+1)}$ ); (II) then we project the less corrupted data  $\mathcal{X} - \mathcal{S}^{(k+1)}$  onto the space of low-multilinear-rank tensors to update the estimate (i.e.,  $\mathcal{L}^{(k+1)}$ ). The key for our algorithm acceleration is using tensor Fiber CUR decomposition for inexact low-multilinear-rank tensor approximation in Step (II), which provides much better computational complexity than the standard HOSVD. Consequently, in Step (I), this inexact approximation allows us to estimate only the outliers that lie in the smaller subtensors and fibers that Fiber CUR decomposition samples. RTCUR is summarized in Algorithm 1. Now, we will discuss the details of our approach and start with Step (II).

### 3.1. Updating $\mathcal{L}$

SVD is the most standard method for low-rank approximation in the matrix setting. Similarly, in the literature, HOSVD is the standard method for low-multilinear-rank approximation under our multilinear rank setting. However, HOSVD is computationally expansive when the problem scale is large. Inspired by the recent development on tensor CUR decomposition [22], we employ Fiber CUR decomposition for accelerated inexact low-multilinear-rank tensor approximations and update the estimate by setting

$$\mathcal{L}^{(k+1)} = \mathcal{R}^{(k+1)} \times_{i=1}^n C_i^{(k+1)} \left( U_i^{(k+1)} \right)^\dagger, \quad (8)$$

---

**Algorithm 1 Robust Tensor CUR (RTCUR)**


---

```

1: Input:  $\mathcal{X} = \mathcal{L}_* + \mathcal{S}_* \in \mathbb{R}^{d_1 \times \dots \times d_n}$ : observed tensor;  $(r_1, \dots, r_n)$ : underlying multilinear rank of  $\mathcal{L}_*$ ;  $\varepsilon$ : targeted precision;  $\zeta^{(0)}, \gamma$ : thresholding parameters;  $\{|I_i|\}_{i=1}^n, \{|J_i|\}_{i=1}^n$ : cardinalities for sample indices.
2: Initialization:  $\mathcal{L}^{(0)} = \mathbf{0}, \mathcal{S}^{(0)} = \mathbf{0}, k = 0$ 
3: Uniformly sample the indices  $\{I_i\}_{i=1}^n, \{J_i\}_{i=1}^n$ 
4: while  $e^{(k)} > \varepsilon$  do //  $e^{(k)}$  is defined in (14)
5:   (Optional) Resample the indices  $\{I_i\}_{i=1}^n, \{J_i\}_{i=1}^n$ 
6:   // Step (I): Updating  $\mathcal{S}$ 
7:    $\zeta^{(k+1)} = \gamma \cdot \zeta^{(k)}$ 
8:    $\mathcal{S}^{(k+1)} = \text{HT}_{\zeta^{(k+1)}}(\mathcal{X} - \mathcal{L}^{(k)})$ 
9:   // Step (II): Updating  $\mathcal{L}$ 
10:   $\mathcal{R}^{(k+1)} = (\mathcal{X} - \mathcal{S}^{(k+1)})(I_1, \dots, I_n)$ 
11:  for  $i = 1, \dots, n$  do
12:     $C_i^{(k+1)} = (\mathcal{R}^{(k+1)})_{(i)}(:, J_i)$ 
13:     $U_i^{(k+1)} = \text{SVD}_{r_i}(C_i^{(k+1)}(I_i, :))$ 
14:  end for
15:   $\mathcal{L}^{(k+1)} = \mathcal{R}^{(k+1)} \times_{i=1}^n C_i^{(k+1)} (U_i^{(k+1)})^\dagger$ 
16:   $k = k + 1$ 
17: end while
18: Output:  $\mathcal{R}^{(k)}, C_i^{(k)}, U_i^{(k)}$  for  $i = 1, \dots, n$ : the estimates of the tensor Fiber CUR decomposition of  $\mathcal{L}_*$ .

```

---

where

$$\begin{aligned}
\mathcal{R}^{(k+1)} &= (\mathcal{X} - \mathcal{S}^{(k+1)})(I_1, \dots, I_n), \\
C_i^{(k+1)} &= (\mathcal{R}^{(k+1)})_{(i)}(:, J_i), \\
U_i^{(k+1)} &= \text{SVD}_{r_i}(C_i^{(k+1)}(I_i, :)),
\end{aligned} \tag{9}$$

for all  $i$ . Therein,  $\text{SVD}_{r_i}$  denotes the truncated rank- $r_i$  matrix singular value decomposition that is used to enforce the low-multilinear-rank constraint. Recall the aforementioned complexity for computing Fiber CUR decomposition is  $\mathcal{O}(r^2 n^2 \log^2(d))$ . Note that there is no need to compute the full  $\mathcal{X} - \mathcal{S}^{(k+1)}$  but merely the sampled subtensors. Similarly, forming the entire  $\mathcal{L}$  is never needed through RTCUR—we only need to keep the decomposed tensor components and form the sampled subtensors of  $\mathcal{L}$  accordingly (see Section 3.3 for details). That is, (8) is provided for explanation but should never be fully computed in an efficient implementation of RTCUR.

### 3.2. Updating $\mathcal{S}$

We consider the simple yet effective hard thresholding operator  $\text{HT}_\zeta$  for outlier estimation, which is defined as:

$$(\text{HT}_\zeta \mathcal{X})_{i_1, \dots, i_n} = \begin{cases} \mathcal{X}_{i_1, \dots, i_n}, & |\mathcal{X}_{i_1, \dots, i_n}| > \zeta; \\ 0, & \text{otherwise.} \end{cases} \tag{10}$$

As shown in [2, 4, 5], with a properly chosen thresholding value,  $\text{HT}_\zeta$  is effectively a projection operator onto the sup-

port of  $\mathcal{S}_*$ . More specifically, we update

$$\mathcal{S}^{(k+1)} = \text{HT}_{\zeta^{(k+1)}}(\mathcal{X} - \mathcal{L}^{(k)}). \tag{11}$$

If  $\zeta^{(k+1)} = \|\mathcal{L}_* - \mathcal{L}^{(k+1)}\|_\infty$  is chosen, then we have  $\text{supp}(\mathcal{S}^{(k+1)}) \subseteq \text{supp}(\mathcal{S}_*)$  and  $\|\mathcal{S}_* - \mathcal{S}^{(k+1)}\|_\infty \leq 2\|\mathcal{L}_* - \mathcal{L}^{(k+1)}\|_\infty$ . Empirically, we find that iteratively decaying thresholding values

$$\zeta^{(k+1)} = \gamma \cdot \zeta^{(k)} \tag{12}$$

provide superb performance with carefully tuned  $\gamma$  and  $\zeta^{(0)}$ . Note that a favorable choice of  $\zeta^{(0)}$  is  $\|\mathcal{L}_*\|_\infty$ , which can be easily accessed/estimated in many applications. The decay factor  $\gamma \in (0, 1)$  should be tuned according to the difficulty of the TRPCA problem, e.g., those problems with higher rank, more dense outliers, or large condition number are harder. For successful reconstruction, the harder problems require larger  $\zeta$  and *vice versa*. We observe that  $\gamma \in [0.6, 0.9]$  generally performs well.

### 3.3. Computational Complexity

As mentioned, the complexity for computing a Fiber CUR decomposition is very low, thus the dominating steps in RTCUR are the hard thresholding operator and/or the tensor/matrix multiplications. We again remark that only the sampled subtensors and fibers are needed when computing (9). Thus, we merely need to estimate the outliers on these subtensors and fibers, and (11) should not be fully executed. Instead, we only compute

$$\begin{aligned}
\mathcal{S}^{(k+1)}(I_1, \dots, I_n) &= \text{HT}_{\zeta^{(k+1)}}((\mathcal{X} - \mathcal{L}^{(k)})(I_1, \dots, I_n)), \\
\mathcal{S}_{(i)}^{(k+1)}(:, J_i) &= \text{HT}_{\zeta^{(k+1)}}((\mathcal{X} - \mathcal{L}^{(k)})_{(i)}(:, J_i)).
\end{aligned} \tag{13}$$

for all  $i$ , which leads to a total complexity of  $\mathcal{O}(r^n \log^n(d) + n^2 dr \log(d))$  in this step. Not only can we save the computational complexity on hard thresholding but also much smaller subtensors of  $\mathcal{L}^{(k)}$  need to be formed in (13). We can form the required subtensors from the stored Fiber CUR components, which is much cheaper than forming and saving the whole  $\mathcal{L}^{(k)}$ . In particular, the total complexity for forming  $\mathcal{L}^{(k)}(I_1, \dots, I_n)$  and  $\{\mathcal{L}_{(i)}^{(k)}(:, J_i)\}_{i=1}^n$  is  $\mathcal{O}(n^2 dr^2 \log^2(d) + n^2 r^{n+1} \log^{n+1}(d))$  and it dominates the overall complexity of RTCUR.

Moreover, for time saving purposes, we may avoid computing the Frobenius norm of the full tensor when computing the relative error for the stopping criterion. In RTCUR, we adjust the relative error formula to be

$$e^{(k)} = \frac{\|\mathcal{E}^{(k)}(I_1, \dots, I_n)\|_F + \sum_{i=1}^n \|\mathcal{E}_{(i)}^{(k)}(:, J_i)\|_F}{\|\mathcal{X}(I_1, \dots, I_n)\|_F + \sum_{i=1}^n \|\mathcal{X}_{(i)}(:, J_i)\|_F} \tag{14}$$

where  $\mathcal{E}^{(k)} = \mathcal{X} - \mathcal{L}^{(k)} - \mathcal{S}^{(k)}$ , so that it does not use any extra subtensor or fiber but only those we already formed.

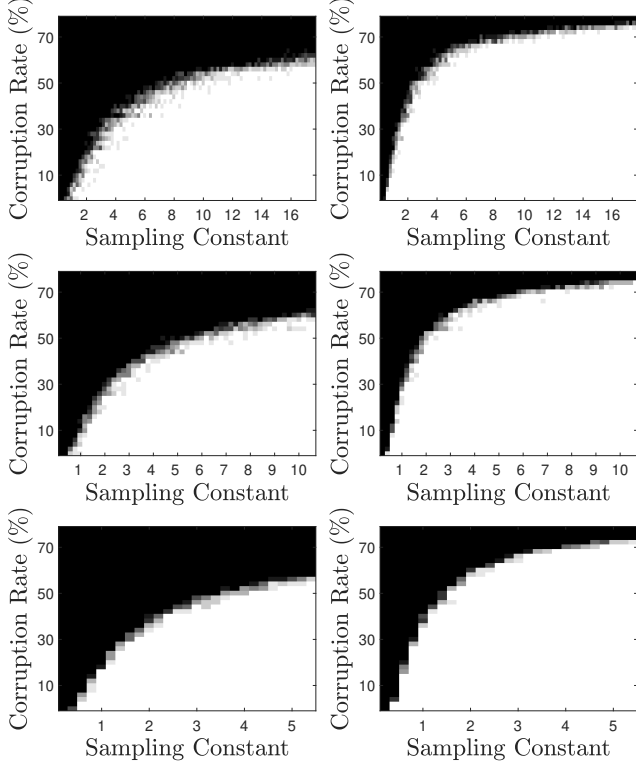


Figure 2: Empirical phase transition in corruption rate  $\alpha$  and sampling constant  $v$ . **Left:** RTCUR-F. **Right:** RTCUR-R. **Top:**  $r = 3$ . **Middle:**  $r = 5$ . **Bottom:**  $r = 10$ .

### 3.4. Two Variants of RTCUR

We consider two variants of RTCUR, namely RTCUR-F and RTCUR-R. RTCUR-F uses fixed sample indices through all iterations while RTCUR-R resamples  $\{I_i\}_{i=1}^n$  and  $\{J_i\}_{i=1}^n$  in every iteration. RTCUR-F requires minimal data accessibility and runs slightly faster. RTCUR-R accesses more data and takes some extra computing; for example, the denominator of (14) has to be recomputed per iteration. On the other hand, accessing more redundant data means RTCUR-R has a better chance to correct any “unlucky” sampling over the iterations, thus we expect that RTCUR-R has superior outlier tolerance than RTCUR-F. We recommend RTCUR-F if the data accessibility or computation ability is limited. Otherwise, if better outlier tolerance is desired and data is easily accessible, then RTCUR-R is recommended.

## 4. Numerical Experiments

In this section, we conduct numerical experiments to verify the empirical performance of RTCUR against the state-of-the-art robust matrix/tensor PCA algorithms: Alternating Direction Method of Multipliers for TRPCA (ADMM) [17], Accelerated Alternating Projections for RPCA (AAP) [4],

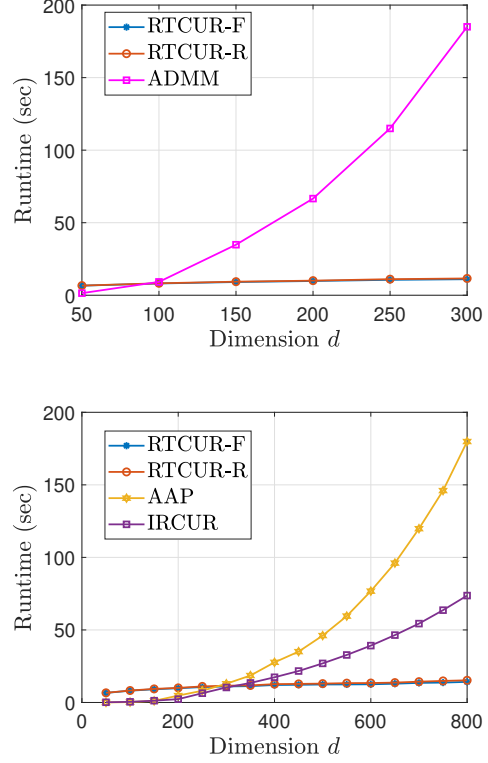


Figure 3: Runtime vs. dimension comparison among RTCUR-F, RTCUR-R, ADMM, AAP, and IRCUR. Multilinear rank (3, 3, 3). The ADMM method proceeded relatively slow for larger tensor so we only test the ADMM runtime for tensor size smaller than 300.

and Iterative Robust CUR for RPCA (IRCUR) [5]. Both proposed variants, RTCUR-F and RTCUR-R, will be evaluated. Section 4.1 contains two synthetic experiments: (i) We study the empirical relation between the outlier tolerance and sample size for RTCUR. (ii) We show the speed advantage of RTCUR compared to the state-of-the-art methods. In Sections 4.2, we successfully apply RTCUR to two real-world problems, color video background subtraction and face modeling.

We utilize the codes of all compared algorithms from the authors’ websites, and the parameters are hand tuned for their best performance. For RTCUR, we sample  $|I_i| = vr_i \log(d_i)$  and  $|J_i| = vr_i \log(\prod_{j \neq i} d_j)$  for all  $i$ , and  $v$  is called the *sampling constant* through this section. All the tests are executed from Matlab R2020a on an Ubuntu workstation with Intel i9-9940X CPU and 128GB RAM.

### 4.1. Synthetic Examples

For the synthetic experiments, we use  $d := d_1 = \dots = d_n$  and  $r := r_1 = \dots = r_n$ . The observed tensor  $\mathcal{X} = \mathcal{L}_\star + \mathcal{S}_\star$ . To generate  $n$ -mode  $\mathcal{L}_\star \in \mathbb{R}^{d \times \dots \times d}$  with multi-



Table 1: Video information and runtime comparison for color video background subtraction task.

	video size	runtime (sec)				
		RTCUR-F	RTCUR-R	ADMM [17]	AAP [4]	IRCUR [5]
<i>Shoppingmall</i>	$256 \times 320 \times 3 \times 1250$	<b>3.53</b>	5.83	783.67	50.38	15.71
<i>Highway</i>	$240 \times 320 \times 3 \times 440$	<b>3.15</b>	5.47	168.55	18.10	3.87
<i>Crossroad</i>	$350 \times 640 \times 3 \times 900$	<b>6.15</b>	13.33	1099.3	97.85	35.47
<i>Port</i>	$480 \times 640 \times 3 \times 1000$	<b>7.22</b>	8.10	3502.8	121.2	11.98
<i>Parking lot</i>	$360 \times 640 \times 3 \times 400$	<b>6.30</b>	8.91	1001.5	43.12	9.02

linear rank  $(r, \dots, r)$ , we take  $\mathcal{L}_* = \mathcal{Y} \times_1 Y_1 \times_2 \dots \times_n Y_n$  where  $\mathcal{Y} \in \mathbb{R}^{r \times \dots \times r}$  and  $\{Y_i \in \mathbb{R}^{d \times r}\}_{i=1}^n$  are Gaussian random tensor and matrices with standard normal entries. To generate the sparse outlier tensor  $\mathcal{S}_*$ , we uniformly sample  $\alpha$  percent entries to be the support of  $\mathcal{S}_*$  and the values of the non-zero entries are uniformly sampled from the interval  $[-\mathbb{E}(|\mathcal{L}_{i_1, \dots, i_n}|), \mathbb{E}(|\mathcal{L}_{i_1, \dots, i_n}|)]$ .

**Phase transition.** We study the empirical relation between the outlier corruption rate  $\alpha$  and sampling constant  $v$  for RTCUR-F and RTCUR-R. The tests are conducted on  $300 \times 300 \times 300$  (i.e.,  $n = 3$  and  $d = 300$ ) problems with  $r = 3, 5$ , or 10. For both the variants, the thresholding parameters are set to be  $\zeta^{(0)} = \|\mathcal{L}\|_\infty$  and  $\gamma = 0.7$ . The stopping condition is  $e^k < 10^{-5}$  and an example is considered successfully solved if  $\|\mathcal{L}_* - \mathcal{L}^{(k)}\|_F / \|\mathcal{L}_*\|_F \leq 10^{-3}$ . For each pair of  $\alpha$  and  $v$ , we generate 10 test examples.

The experimental results are summarized in Figure 2, where a white pixel means all 10 test examples are successfully solved under the corresponding problem parameter setting and a black pixel means all 10 test cases fail. As expected, one can see RTCUR-R has slightly better performance than RTCUR-F and smaller  $r$  tolerates more outliers since a larger  $r$  leads to a harder problem. Moreover, taking larger  $v$  clearly provides better outlier tolerance. However, larger  $v$  means RTCUR needs to sample larger subtensors and more fibers, which of course leads to more computational time. According to this experiment, we recommend  $v \in [3, 5]$  to balance between outlier tolerance and speed.

**Computational efficiency.** In this section, we compare the computational efficiency between RTCUR and the state-of-the-art tensor/matrix RPCA algorithms. For matrix methods, we first unfold a  $n$ -mode  $d \times \dots \times d$  tensor to a  $d \times d^{n-1}$  matrix, then solve the matrix RPCA problem with rank  $r$ . For all tests, we add 10% corruption and set parameters  $v = 3$ ,  $\zeta^{(0)} = \|\mathcal{L}\|_\infty$ ,  $\gamma = 0.7$  for both RTCUR-F and RTCUR-R. The reported runtime is averaged over 10 trials.

In Figure 3, we consider the problem of 3-mode TRPCA with varying dimension  $d$  and compare the total runtime (all methods halt when  $e^{(k)} < 10^{-5}$ ). One can see both variants of RTCUR are substantially faster than the compared algorithms when  $d$  is large.

## 4.2. Color Video Background Subtraction

We apply RTCUR on the task of color video background subtraction. We consider two color video datasets with static backgrounds: *Shoppingmall* [29], *Highway* [30], *Crossroad* [30], *Port* [30], and *Parking Lot* [31]. Since a monochromatic frame usually does not have low rank structure [32], we vectorize each color channel of each frame into a vector and construct a  $(\text{height} \times \text{width}) \times 3 \times \text{frames}$  tensor. The targeted multilinear rank is  $\mathbf{r} = (3, 3, 3)$  for both videos. For those matrix algorithms, we unfold the tensor to a  $(\text{height} \times \text{width}) \times (3 \times \text{frames})$  matrix and rank 3 is used. We set RTCUR parameters  $v = 2$ ,  $\zeta^{(0)} = 255$ ,  $\gamma = 0.7$  in this experiment.

The test results along with video size information are summarized in Table 1. In addition, we provide some selected visual results in Figure 4. All tested methods produce visually desirable backgrounds while RTCUR-F spends least time in the tests of all the three videos.

## 5. Conclusion and Future Work

This paper presents a highly efficient algorithm RTCUR for large-scale TRPCA problems. RTCUR is developed by introducing a novel inexact low-multilinear-rank tensor approximation via Fiber CUR decomposition, whose structure provides a significantly reduced computational complexity. Specifically speaking, RTCUR has per iteration computational complexity as low as  $\mathcal{O}(n^2 d r^2 \log^2(d) + n^2 r^{n+1} \log^{n+1}(d))$  flops. Numerical experiments show that RTCUR is superior to other state-of-the-art tensor/matrix RPCA algorithms, on both synthetic and real-world datasets.

There are four lines of future research work. First, it will be important to investigate a theoretical convergence guarantee of RTCUR. Second, extending RTCUR to the partially observed setting (i.e., when only a small random portion of the tensor is observed) is another future direction. Third, we will study the stability of RTCUR with additive dense noise since outliers often present with small dense noise in real-world applications. Last, there are exist tensor PCA studies focusing on the online tensor analysis in order to alleviate the heavy memory cost [33–35], we will leave this topic to be our future work.

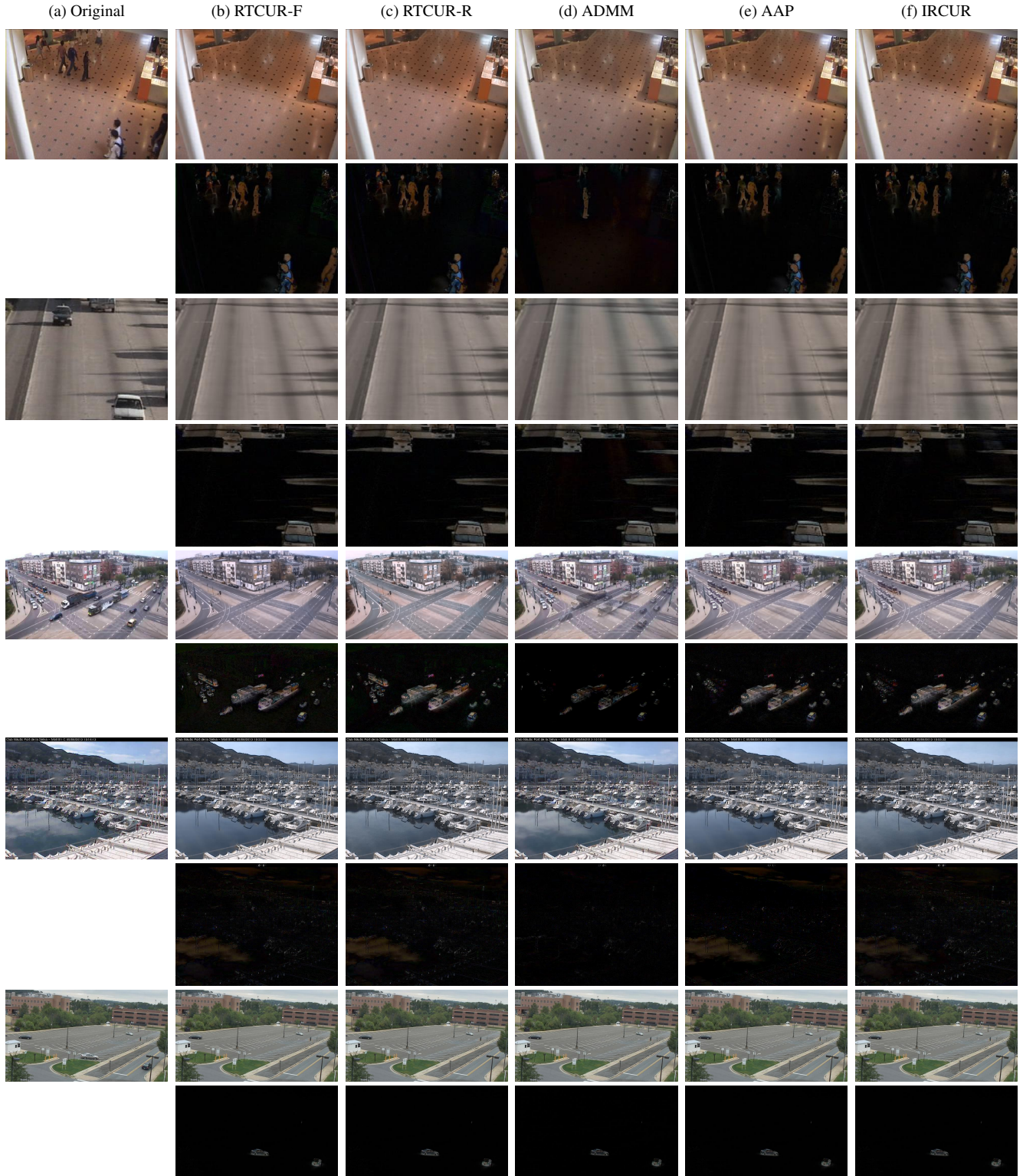


Figure 4: Visual results for color video background subtraction. The **first two rows** are separated backgrounds and foregrounds corresponding to a frame from *Shoppingmall*, the **3rd and 4th rows** are separated backgrounds and foregrounds corresponding to a frame from *Highway*, the **5th and 6th** are separated backgrounds and foregrounds corresponding to a frame from *Crossroad*, **7th and 8th** correspond to a frame from *Port*, and the **last two rows** correspond to a frame from *Parking lot*, except the first column which is the original frame.

## References

- [1] E. J. Candès, X. Li, Y. Ma, and J. Wright, “Robust principal component analysis?,” *Journal of the ACM (JACM)*, vol. 58, no. 3, pp. 1–37, 2011.
- [2] P. Netrapalli, N. U N, S. Sanghavi, A. Anandkumar, and P. Jain, “Non-convex robust PCA,” in *Advances in Neural Information Processing Systems* (Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K. Q. Weinberger, eds.), vol. 27, Curran Associates, Inc., 2014.
- [3] T. Bouwmans, S. Javed, H. Zhang, Z. Lin, and R. Otazo, “On the applications of robust PCA in image and video processing,” *Proceedings of the IEEE*, vol. 106, no. 8, pp. 1427–1457, 2018.
- [4] H. Cai, J.-F. Cai, and K. Wei, “Accelerated alternating projections for robust principal component analysis,” *The Journal of Machine Learning Research*, vol. 20, no. 1, pp. 685–717, 2019.
- [5] H. Cai, K. Hamm, L. Huang, J. Li, and T. Wang, “Rapid robust principal component analysis: CUR accelerated inexact low rank estimation,” *IEEE Signal Processing Letters*, vol. 28, pp. 116–120, 2020.
- [6] H. Cai, K. Hamm, L. Huang, and D. Needell, “Robust CUR decomposition: Theory and imaging applications,” *arXiv preprint arXiv:2101.05231*, 2021.
- [7] J. Wright, A. Y. Yang, A. Ganesh, S. S. Sastry, and Y. Ma, “Robust face recognition via sparse representation,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 31, no. 2, pp. 210–227, 2008.
- [8] Y. Hu, J.-X. Liu, Y.-L. Gao, and J. Shang, “DSTPCA: Double-sparse constrained tensor principal component analysis method for feature selection,” *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 2019.
- [9] H. Cai, J.-F. Cai, T. Wang, and G. Yin, “Accelerated structured alternating projections for robust spectrally sparse signal recovery,” *IEEE Transactions on Signal Processing*, vol. 69, pp. 809–821, 2021.
- [10] L. Li, W. Huang, I. Y.-H. Gu, and Q. Tian, “Statistical modeling of complex backgrounds for foreground object detection,” *IEEE Transactions on Image Processing*, vol. 13, no. 11, pp. 1459–1472, 2004.
- [11] B. Huang, C. Mu, D. Goldfarb, and J. Wright, “Provable low-rank tensor recovery,” *Optimization-Online*, vol. 4252, no. 2, pp. 455–500, 2014.
- [12] Q. Gu, H. Gui, and J. Han, “Robust tensor decomposition with gross corruption,” *Advances in Neural Information Processing Systems*, vol. 27, pp. 1422–1430, 2014.
- [13] S. E. Sofuoglu and S. Aviyente, “A two-stage approach to robust tensor decomposition,” in *2018 IEEE Statistical Signal Processing Workshop (SSP)*, pp. 831–835, IEEE, 2018.
- [14] Y. Hu and D. B. Work, “Robust tensor recovery with fiber outliers for traffic events,” *ACM Transactions on Knowledge Discovery from Data (TKDD)*, vol. 15, no. 1, pp. 1–27, 2020.
- [15] Z. Zhang, G. Ely, S. Aeron, N. Hao, and M. Kilmer, “Novel methods for multilinear data completion and de-noising based on tensor-SVD,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3842–3849, 2014.
- [16] Y. Liu, L. Chen, and C. Zhu, “Improved robust tensor principal component analysis via low-rank core matrix,” *IEEE Journal of Selected Topics in Signal Processing*, vol. 12, no. 6, pp. 1378–1389, 2018.
- [17] C. Lu, J. Feng, Y. Chen, W. Liu, Z. Lin, and S. Yan, “Tensor robust principal component analysis with a new tensor nuclear norm,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 42, no. 4, pp. 925–938, 2019.
- [18] F. L. Hitchcock, “The expression of a tensor or a polyadic as a sum of products,” *Journal of Mathematical Physics*, vol. 6, no. 1-4, pp. 164–189, 1927.
- [19] A. Anandkumar, P. Jain, Y. Shi, and U. N. Niranjan, “Tensor vs. matrix methods: Robust tensor decomposition under block sparse perturbations,” in *Artificial Intelligence and Statistics*, pp. 268–276, PMLR, 2016.
- [20] F. L. Hitchcock, “Multiple invariants and generalized rank of a  $p$ -way matrix or tensor,” *Journal of Mathematical Physics*, vol. 7, no. 1-4, pp. 39–79, 1928.
- [21] D. Goldfarb and Z. Qin, “Robust low-rank tensor recovery: Models and algorithms,” *SIAM Journal on Matrix Analysis and Applications*, vol. 35, no. 1, pp. 225–253, 2014.
- [22] H. Cai, K. Hamm, L. Huang, and D. Needell, “Mode-wise tensor decompositions: Multi-dimensional generalizations of CUR decompositions,” *The Journal of Machine Learning Research*, vol. 22, no. 185, pp. 1–36, 2021.
- [23] L. De Lathauwer, B. De Moor, and J. Vandewalle, “On the best rank-1 and rank- $(r_1, \dots, r_n)$  approximation of higher-order tensors,” *SIAM Journal on Matrix Analysis and Applications*, vol. 21, no. 4, pp. 1324–1342, 2000.
- [24] S. A. Goreňov, N. L. Zamarashkin, and E. E. Tyrtshnikov, “Pseudo-skeleton approximations,” *Doklady Akademii Nauk*, vol. 343, no. 2, pp. 151–152, 1995.
- [25] K. Hamm and L. Huang, “Perspectives on CUR decompositions,” *Applied and Computational Harmonic Analysis*, vol. 48, no. 3, pp. 1088–1099, 2020.
- [26] M. W. Mahoney, M. Maggioni, and P. Drineas, “Tensor-CUR decompositions for tensor-based data,” *SIAM Journal on Matrix Analysis and Applications*, vol. 30, no. 3, pp. 957–987, 2008.
- [27] C. F. Caiafa and A. Cichocki, “Generalizing the column–row matrix decomposition to multi-way arrays,” *Linear Algebra and its Applications*, vol. 433, no. 3, pp. 557–573, 2010.
- [28] K. Hamm and L. Huang, “Stability of sampling for CUR decompositions,” *Foundations of Data Science*, vol. 2, no. 2, p. 83, 2020.
- [29] E. L., “Foreground detection in video sequences with probabilistic self-organizing maps.”



- [30] T. Bouwmans, L. Maddalena, and A. Petrosino, "Scene background initialization: A taxonomy," *Pattern Recognition Letters*, vol. 96, pp. 3–11, 2017.
- [31] S. Oh, A. Hoogs, A. Perera, N. Cuntoor, C.-C. Chen, J. T. Lee, S. Mukherjee, J. Aggarwal, H. Lee, L. Davis, *et al.*, "A large-scale benchmark dataset for event recognition in surveillance video," in *CVPR 2011*, pp. 3153–3160, IEEE, 2011.
- [32] Z. Chao, L. Huang, and D. Needell, "Tensor completion through total variation with initialization from weighted hotsvd," in *2020 Information Theory and Applications Workshop (ITA)*, pp. 1–8, IEEE, 2020.
- [33] A. Sobral, C. G. Baker, T. Bouwmans, and E.-h. Zahzah, "Incremental and multi-feature tensor subspace learning applied for background modeling and subtraction," in *International Conference Image Analysis and Recognition*, pp. 94–103, Springer, 2014.
- [34] P. Li, J. Feng, X. Jin, L. Zhang, X. Xu, and S. Yan, "Online robust low-rank tensor modeling for streaming data analysis," *IEEE transactions on neural networks and learning systems*, vol. 30, no. 4, pp. 1061–1075, 2018.
- [35] J. Bin, M. Kang, and Z. Liu, "Gpu-accelerated tensor decomposition for moving object detection from multimodal imaging," in *2020 IEEE Sensors*, pp. 1–4, IEEE, 2020.