RE-LL1: AN EFFECTIVE REGULARIZED $(L, L, 1)$-TENSOR DECOMPOSITION METHOD FOR VIDEO BACKGROUND MODELING AND FOREGROUND SEPARATION

Nguyen Quy Dang**, Thanh Trung Le***, Nguyen Linh Trung***, Karim Abed-Meraim*†

* VNU University of Engineering and Technology, Vietnam National University, Hanoi, Vietnam
† PRISME Laboratory, University of Orleans, France

ABSTRACT

In this paper, we proposed an efficient regularized $(L, L, 1)$-decomposition method, called Re-LL1, to factorize a time-series tensor into block low-rank terms and sparse components. Re-LL1 integrates three types of regularization techniques, including nuclear norm for compactness and low rank, generalized ℓ_p-norm for sparsity, and Tikhonov regularization for temporal smoothness. To solve the model efficiently, we develop an elegant block-wise augmented Lagrangian method with Anderson acceleration. Experiments on real-world surveillance video datasets demonstrate that Re-LL1 achieves robust background–foreground separation and converges faster than state-of-the-art methods.

Index Terms— Tensor decomposition, LL1 decomposition, data imperfection, data corruption, regularization, video background and foreground separation.

1. INTRODUCTION

Tensor decomposition (TD) techniques have been increasingly applied across a wide range of signal processing and data science domains, demonstrating their effectiveness in extracting meaningful structure from multidimensional signals and datasets [1–4]. Notable applications of TD include biomedical signal processing, hyperspectral imaging, video background-foreground separation, and anomaly detection, where the goal is to robustly separate structured data from sparse corruptions and anomalies [5].

Among TD models, the block term decomposition (BTD) framework, particularly LL1 model, stands out due to its ability to factorize tensors into a sum of block low-rank components [6]. Under the LL1 format, each block is expressed as the outer product of a low-rank matrix and a coefficient vector. This model can be viewed as an extension of the most widely used CP/PARAFAC tensor decomposition. In this study, we aim to develop effective methods for LL1 decomposition in the presence of data imperfection, with a specific focus on the application of video background and foreground separation.

Data imperfection remains a critical challenge in tensor decomposition, often leading to degraded performance, especially in real-world applications [3]. For instance, in video processing and analysis, sparse outliers typically correspond to moving objects or sudden environmental changes that traditional TD methods struggle to handle effectively. To ensure reliable performance, robust approaches that explicitly account for such corruptions are essential.

Most existing methods for LL1 and block term decomposition (BTD) rely on frameworks, including alternating least squares (ALS) [7], block coordinate descent (BCD) [8], nonlinear least squares (NLS) [8], Bayesian approach [9], and their variants [10–13]. However, in the presence of data imperfection, these techniques often suffer from slow convergence and low estimation accuracy (See Fig. 2 for an example). To address these limitations, some robust methods have been introduced. For example, the alternating group-lasso (AGL) and hierarchical iterative reweighted least squares (IRLS) approaches employ mixed $\ell_{1,2}$-norm penalties to promote sparsity and enhance robustness by automatically pruning irrelevant blocks and columns [10]. Extending the IRLS framework, the work in [14] proposed a robust BTD-based denoising method capable of handling sparse outliers by reweighting the ℓ_1-norm of sparse components, particularly in hyperspectral imaging. Similarly, [15] presented a spectral-spatial ℓ_0-norm regularized BTD method for hyperspectral image denoising. Despite their theoretical strengths, these methods are designed for a specific type of data (i.e., hyperspectral images) and hence limit their practical values in other applications.

In this study, we contribute to the tensor literature by introducing Re-LL1, a novel and effective LL1 decomposition method with regularizations. We design a sparsity-aware LL1/BTD model that can enforce the compactness of low-rank components, identify sparse corruptions, and exploit the inherent smoothness of tensor data. To this end, we incorporate a set of carefully designed regularization techniques that fully exploit the structural properties of LL1 factors. To solve the resulting optimization problem, we develop an efficient block-wise augmented Lagrangian method enhanced with Anderson acceleration, ensuring both fast convergence and computational stability. Finally, we validate the proposed method through real-world video background–foreground separation tasks, where Re-LL1 consistently outperforms existing state-of-the-art approaches.

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

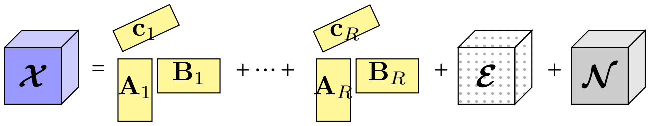


Fig. 1: Regularized LL1 tensor decomposition: Decompose a tensor $\mathcal{X}$ into block low-rank terms plus a sparse component $\mathcal{E}$ and an additive Gaussian noise $\mathcal{N}$.

2. REGULARIZED LL1 TENSOR DECOMPOSITION

In this work, we consider a time-series tensor $\mathcal{X} \in \mathbb{R}^{I\times J\times K}$ which is corrupted by a sparse component $\mathcal{E}$ and an additive Gaussian noise $\mathcal{N}$, see Fig.1 for an illustration. The LL1 decomposition of $\mathcal{X}$ can be formulated as follows

$$\mathcal{X} = \sum_{r=1}^{R} \big(\mathbf{A}_r \mathbf{B}_r^\top\big) \circ \mathbf{c}_r + \mathcal{E} + \mathcal{N}, \tag{1}$$

where " $\circ$ " denotes the outer product, matrices $\mathbf{A}_r \in \mathbb{R}^{I\times L}$ and $\mathbf{B}_r \in \mathbb{R}^{J\times L}$ form a rank-L matrix $\mathbf{H}_r = \mathbf{A}_r\mathbf{B}_r^\top \in \mathbb{R}^{I\times J}$, and $\mathbf{c}_r \in \mathbb{R}^{K\times 1}$ is a non-zero coefficent vector [6]. To satisfy the conditions for essential uniqueness of LL1, we assume that $\{\mathbf{A}_r, \mathbf{B}_r\}_{r=1}^R$ are full column rank matrices and $\mathbf{C} = [\mathbf{c}_1, \mathbf{c}_2, \dots, \mathbf{c}_R]$ has no proportional columns.

Due to the presence of the sparse component $\mathcal{E}$, standard optimization approaches for LL1/BTD (e.g., ALS [7] and NLS [16]) become ineffective for decomposing $\mathcal{X}$ in (1). To address this, we incorporate regularization techniques that promote sparsity, enforce smoothness, and encourage compactness, thereby improving the accuracy and robustness of the decomposition, as follows

$$\begin{aligned} \underset{\{\mathbf{A}_r,\mathbf{B}_r,\mathbf{c}_r\}_{r=1}^R,\mathcal{E}}{\operatorname{argmin}} \; & \frac{1}{2}\Big\| \mathcal{X} - \sum_{r=1}^{R}\big(\mathbf{A}_r\mathbf{B}_r^\top \circ \mathbf{c}_r\big) - \mathcal{E}\Big\|_F^2 \\ & + \lambda_1 \mathcal{R}_S(\mathcal{E}) + \lambda_2 \sum_{r=1}^{R} \mathcal{R}_T(\mathbf{c}_r) + \lambda_3 \sum_{r=1}^{R}\mathcal{R}_L(\mathbf{A}_r, \mathbf{B}_r). \end{aligned} \tag{2}$$

Here, the first term, based on Frobenius norm, aims to minimize the residual error between the observed data and its estimate, thereby enforcing data fidelity. To promote the sparsity on $\mathcal{E}$, we propose to use the following ℓ_p-norm regularization

$$\mathcal{R}_S(\mathcal{E}) = \Big(\sum_{i,j,k} \big|\mathcal{E}_{i,j,k}\big|^p\Big)^{1/p}, \tag{3}$$

for $0 \le p \le 1$. When $p = 1$, (3) reduces to the most widely-used ℓ_1-norm regularization. When $p < 1$, it becomes nonconvex and more forcefully encouraging zero-valued entries. Accordingly, (2) is well-suited for capturing sharp, localized anomalies and outliers in datasets such as moving objects in video surveillance sequences.

To ensure the inherent smoothness and temporal correlation of times series, we regularize $\mathbf{c}_r, r = 1, 2, \dots, R$ using

$$\mathcal{R}_T(\mathbf{c}_r) = \|\mathbf{c}_r\|_{\mathbf{L}_G}^2 := \mathbf{c}_r^\top \mathbf{L}_G \mathbf{c}_r, \tag{4}$$

where $\mathbf{L}_G \in \mathbb{R}^{K\times K}$ is a positive semidefinte matrix. In the context of video processing, $\mathbf{L}_G$ can be chosen as a graph Laplacian that encodes temporal and/or structural adjacency. In fact, (4) can be interpreted as a form of Tikhonov regularization. Since $\mathbf{L}_G \succeq \mathbf{0}$, there always exists a matrix $\mathbf{\Gamma}$ such that $\mathbf{\Gamma}\mathbf{\Gamma}^\top = \mathbf{L}_G$ allowing us to express the regularizer as $\mathcal{R}_T(\mathbf{c}_r) = \|\mathbf{\Gamma}\mathbf{c}_r\|_2^2$. Particularly when $\mathbf{L}$ is an identify matrix, it becomes the standard ℓ_2-norm regularization that penalizes large coefficients and avoids ill-posed problems.

To improve the compactness and low rank of block term factors, we apply the following nuclear norm regularization

$$\mathcal{R}_L(\mathbf{A}_r, \mathbf{B}_r) = \big\|\mathbf{A}_r\mathbf{B}_r^\top\big\|_*, \tag{5}$$

where $(\cdot)_*$ denotes the nuclear norm which is the sum of singular values of a matrix. This is because the nuclear norm is the tightest convex envelope of the rank function of a matrix. The following lemma provides an alternative formulation of (5), which is utilized by our Re-LL1 method to be presented in the next section.

Lemma 1 (Lemma 6 in [17]). *For any matrix $\mathbf{H} \in \mathbb{R}^{m\times n}$, its nuclear norm can be obtained by*

$$\|\mathbf{H}\|_* = \min_{\mathbf{A}\mathbf{B}^\top = \mathbf{H}} \frac{1}{2}\big(\|\mathbf{A}\|_F^2 + \|\mathbf{B}\|_F^2\big),$$

whose global minimizer is achieved at $\hat{\mathbf{A}} = \mathbf{U}\sqrt{\mathbf{\Sigma}}$ and $\hat{\mathbf{B}} = \mathbf{V}\sqrt{\mathbf{\Sigma}}$ where $[\mathbf{U}, \mathbf{\Sigma}, \mathbf{V}] = \mathrm{SVD}(\mathbf{H})$ is the singular value decomposition (SVD) of $\mathbf{H}$.

Proof. See Appendix A.5 in [17].

3. PROPOSED METHOD

This section presents an effective method for regularized LL1 decomposition based on the augmented Lagrangian approach [18].

In this work, we introduce a new dual tensor $\mathcal{Y}$ with a penalty parameter $\mu > 0$ and reformulate the main objective function (2) as the following augmented Lagrangian

$$\begin{aligned} \mathcal{L}_\mu(\{\mathbf{A}_r, \mathbf{B}_r, \mathbf{c}_r\}_{r=1}^R, \mathcal{E}\,; \mathcal{Y}) = & \\ + \lambda_1 \mathcal{R}_S(\mathcal{E}) + \lambda_2 \sum_{r=1}^{R}\mathcal{R}_T(\mathbf{c}_r) + \lambda_3 \sum_{r=1}^{R} \mathcal{R}_L(\mathbf{A}_r, \mathbf{B}_r) & \\ + \big\langle \mathcal{Y},\, \mathcal{X} - \mathcal{L} - \mathcal{E}\big\rangle + \frac{\mu + 1}{2}\big\|\mathcal{X} - \mathcal{L} - \mathcal{E}\big\|_F^2, & \end{aligned} \tag{6}$$

where $\mathcal{L} = \sum_{r=1}^R \big(\mathbf{A}_r\mathbf{B}_r^\top\big) \circ \mathbf{c}_r$. Thanks to the duality theory, we can find the stationary point of (6) by using the following iteration procedure: minimize $\mathcal{L}_\mu(\cdot)$ with respect to each block variable ($\mathbf{A}_r, \mathbf{B}_r, \mathbf{c}_r$ and $\mathcal{E}$), while keeping the others fixed, and then update the dual variable $\mathcal{Y}$. In particular, at the $(\ell + 1)$-th iteration, we perform three main stages:

Stage 1: We update the underlying block low-rank components by minimizing the following problem

$$\begin{aligned} \underset{\{\mathbf{A}_r,\mathbf{B}_r,\mathbf{c}_r\}_{r=1}^R}{\operatorname{argmin}} \; & \frac{\mu+1}{2}\Big\|\mathcal{Z}^{(\ell)} - \sum_{r=1}^{R}(\mathbf{A}_r\mathbf{B}_r^\top)\circ \mathbf{c}_r\Big\|_F^2 \\ & + \lambda_2 \sum_{r=1}^{R}\mathcal{R}_T(\mathbf{c}_r) + \lambda_3 \sum_{r=1}^{R}\mathcal{R}_L(\mathbf{A}_r, \mathbf{B}_r), \end{aligned} \tag{7}$$

with $\mathcal{Z}^{(\ell)} = \mathcal{X} - \mathcal{E}^{(\ell)} + \frac{1}{\mu}\mathcal{Y}^{(\ell)}$, where $\mathcal{E}^{(\ell)}$ and $\mathcal{Y}^{(\ell)}$ are the estimates of $\mathcal{E}$ and $\mathcal{Y}$ from the previous iteration. We solve (7) by iteratively cycling through $r = 1, 2, \ldots, R$ as follows

$$\boldsymbol{\mathcal{Z}}^{(\bar{r})} = \boldsymbol{\mathcal{Z}}^{(\ell)} - \sum_{s \neq r} \big(\mathbf{A}_s^{(\ell)} (\mathbf{B}_s^{(\ell)})^\top\big) \circ \mathbf{c}_s^{(\ell)}, \tag{8}$$

$$\mathbf{c}_r^{(\ell+1)} = \underset{\mathbf{c}}{\operatorname{argmin}} \frac{\mu+1}{2} \sum_{k=1}^{K} \big\| \boldsymbol{\mathcal{Z}}^{(\bar{r})}_{:,:,k} - c_k \big(\mathbf{A}_r^{(\ell)} (\mathbf{B}_r^{(\ell)})^\top\big)\big\|_F^2 + \lambda_2 \|\mathbf{\Gamma}\, \mathbf{c}\|_2^2, \tag{9}$$

$$\big\{\mathbf{A}_r^{(\ell+1)}, \mathbf{B}_r^{(\ell+1)}\big\} = \underset{\mathbf{A},\mathbf{B}}{\operatorname{argmin}} \frac{\mu+1}{2} \big\|\mathbf{M}^{(\ell)} - \mathbf{A}_r \mathbf{B}_r^\top\big\|_F^2 + \frac{\lambda_3}{2}\big(\|\mathbf{A}\|_F^2 + \|\mathbf{B}\|_F^2\big), \tag{10}$$

where $\mathbf{M}^{(\ell)} = \frac{1}{\omega^{(\ell)}} \sum_k c_k^{(\ell+1)} \boldsymbol{\mathcal{Z}}^{(\bar{r})}_{:,:,k}$ and $\omega^{(\ell)} = \sum_k (c_k^{(\ell+1)})^2$. Here, the k-th element $c_k^{(\ell+1)}$ of the solution $\mathbf{c}_r^{(\ell+1)}$ of (9) can be determined as

$$c_k^{(\ell+1)} = \frac{\big\langle \boldsymbol{\mathcal{Z}}^{(\bar{r})}_{:,:,k},\, \mathbf{A}_r^{(\ell)} (\mathbf{B}_r^{(\ell)})^\top \big\rangle}{\big\|\mathbf{A}_r^{(\ell)} (\mathbf{B}_r^{(\ell)})^\top\big\|_F^2 + \frac{2\lambda_2\, \Gamma_k^2}{\mu+1}}, \quad \Gamma_k = \sum_j \mathbf{\Gamma}(j,k). \tag{11}$$

Leveraging Lemma 1, the solution of (10) can be obtained by performing the truncated singular value decomposition (SVD) of $\mathbf{M}^{(\ell)}$, followed by a $\frac{\lambda_3}{\mu+1}$-thresholding applied on its top L singular values, as follows

$$\big\{\mathbf{U}^{(\ell)}, \mathbf{\Sigma}^{(\ell)}, \mathbf{V}^{(\ell)}\big\} = \operatorname{SVD}\big(\mathbf{M}^{(\ell)}, L\big), \tag{12}$$

$$\widetilde{\mathbf{\Sigma}}^{(\ell)}(l,l) = \max\Big(\mathbf{\Sigma}^{(\ell)}(l,l) - \tfrac{\lambda_3}{\mu+1}, 0\Big),\ 1 \leq l \leq L, \tag{13}$$

$$\mathbf{A}_r^{(\ell+1)} = \mathbf{U}^{(\ell)} \sqrt{\widetilde{\mathbf{\Sigma}}^{(\ell)}}, \qquad \mathbf{B}_r^{(\ell+1)} = \mathbf{V} \sqrt{\widetilde{\mathbf{\Sigma}}^{(\ell)}}. \tag{14}$$

Stage 2: We then estimate the sparse component $\boldsymbol{\mathcal{E}}^{(\ell+1)}$ by

$$\boldsymbol{\mathcal{E}}^{(\ell+1)} = \underset{\boldsymbol{\mathcal{E}}}{\operatorname{argmin}} \frac{\mu+1}{2} \big\|\boldsymbol{\mathcal{X}} - \boldsymbol{\mathcal{L}}^{(\ell+1)} + \tfrac{1}{\mu}\boldsymbol{\mathcal{Y}}^{(\ell)} - \boldsymbol{\mathcal{E}}\big\|_F^2 + \lambda_1 \mathcal{R}_s(\boldsymbol{\mathcal{E}}), \tag{15}$$

where $\boldsymbol{\mathcal{L}}^{(\ell+1)} = \sum_{r=1}^{R} \big(\mathbf{A}_r^{(\ell+1)} (\mathbf{B}_r^{(\ell+1)})^\top\big) \circ \mathbf{c}_r^{(\ell+1)}$. Here, we apply the proximal operator of the function $\mathcal{R}_s(\cdot)$ as

$$\boldsymbol{\mathcal{E}}^{(\ell+1)} = \operatorname{prox}_{\frac{\lambda_1}{\mu+1}, \mathcal{R}_s}\big(\boldsymbol{\mathcal{X}} - \boldsymbol{\mathcal{L}}^{(\ell+1)} + \tfrac{1}{\mu}\boldsymbol{\mathcal{Y}}^{(\ell)}\big). \tag{16}$$

For example, when $p = 1$ or $p = 0$, the well-known soft- or hard-thresholding operators can be used, respectively. In the case of $0 < p < 1$, the proximal operator proposed in [19] can be adopted.

Stage 3: We update the dual variable $\boldsymbol{\mathcal{Y}}^{(\ell+1)}$ by maximizing the Lagrangian function (6) with respect to $\boldsymbol{\mathcal{Y}}$, yielding

$$\boldsymbol{\mathcal{Y}}^{(\ell+1)} = \boldsymbol{\mathcal{Y}}^{(\ell)} + (\mu+1)\big(\boldsymbol{\mathcal{X}} - \boldsymbol{\mathcal{L}}^{(\ell+1)} - \boldsymbol{\mathcal{E}}^{(\ell+1)}\big). \tag{17}$$

Stopping criterion: The iteration procedure stops when $\|\boldsymbol{\mathcal{X}} - \boldsymbol{\mathcal{L}} - \boldsymbol{\mathcal{E}}\|_F / \|\boldsymbol{\mathcal{X}}\|_F <$ tol with a predefined tolerance error tol or when the maximum number of iterations is reached.

Anderson Mixing: To accelerate the iteration procedure, we apply the Anderson mixing method [20]. Instead of directly using the closed-form update (11) at every iteration, we extrapolate a new estimate of $\mathbf{c}_r^{(\ell+1)}$ from the last m iterations and residuals. In particular, we construct the following residual matrix $\mathbf{G}_r^{(\ell)} = \big[\mathbf{g}_r^{(\ell-m+1)}, \ldots, \mathbf{g}_r^{(\ell-1)}, \mathbf{g}_r^{(\ell)}\big]$, where $\mathbf{g}_r^{(k)} = \mathbf{c}_r^{(k)} - \mathbf{c}_r^{(\ell+1)}$ with $k \leq \ell$. Then, we find the weight vector $\mathbf{w}_r^{(\ell)} = \mathbf{z}_r^{(\ell)} / (\mathbf{1}^\top \mathbf{z}_r^{(\ell)})$ where $\mathbf{z}_r^{(\ell)}$ is obtained by solving the following equation

$$\big[(\mathbf{G}_r^{(\ell)})^\top \mathbf{G}_r^{(\ell)} + \epsilon \mathbf{I}\big] \mathbf{z}_r^{(\ell)} = \mathbf{1}, \tag{18}$$

with a damping parameter $\epsilon > 0$. Finally, with a mixing step h, we re-update $\mathbf{c}_r^{(\ell+1)}$ as follows

$$\mathbf{c}_r^{(\ell+1)} = \textstyle\sum_{i=0}^{m-1} \mathbf{w}_r^{(\ell)}(i) \big(\mathbf{c}_r^{(\ell-i)} - h\, \mathbf{g}_r^{(\ell-i)}\big). \tag{19}$$

To improve efficiency, we apply three enhancements. First, we use a capped and adaptive history strategy, dynamically adjusting $m \leq m_{\max}$ based on recent residuals. Second, we perform a line search at each iteration to adaptively select the damping parameter ϵ. Third, we incrementally update the Cholesky factorization of $(\mathbf{G}_r^{(\ell)})^\top \mathbf{G}_r^{(\ell)} + \epsilon \mathbf{I}$, reducing the cost of the least-squares solve in (18).

4. NUMERICAL EXPERIMENTS

4.1. LL1 Tensor Decomposition

First, we evaluate the performance of Re-LL1 for robust LL1 decomposition with both simulated data and real video data.[1] Here, we generate a synthetic tensor $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{40 \times 50 \times 60}$, based on the data model (1). The tensor is constructed using six random factors $\{\mathbf{A}_r, \mathbf{B}_r\}_{r=1}^{6}$ where $\mathbf{A}_r \in \mathbb{R}^{40 \times 16}$, $\mathbf{B}_r \in \mathbb{R}^{50 \times 16}$, along with the temporal factor $\mathbf{C} \in \mathbb{R}^{60 \times 6}$. All factor matrices are drawn from Gaussian distributions with zero mean and unit variance. Sparse outliers are introduced in the tensor $\boldsymbol{\mathcal{E}}$, affecting 10% of the total entries. The magnitudes of these outliers are scaled to five times the maximum entry of the tensor. The entries of $\boldsymbol{\mathcal{N}}$ are generated from a Gaussian distribution with a mean of 0 and a variance of 0.1.

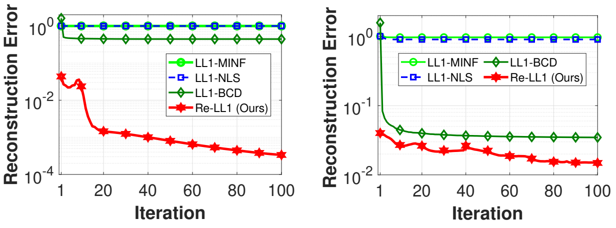
In the latter case, we use the `Office` surveillance video dataset which captures people entering and exiting an office room. This video is represented as a 3rd-order tensor of size $240 \times 360 \times 300$, where 240×260 corresponds to the spatial dimensions of each video frame and 300 denotes the number of frames considered. The number of block terms and their rank were set to $R = 6$ and $L = 16$, respectively.

We compare the performance of Re-LL1 with three well-known LL1 decomposition methods, including LL1-MINF, LL1-NLS from TensorLab [8], and LL1-BCD [7]. All methods are initialized at random and evaluated using the reconstruction error (RE) metric, which is defined as $\text{RE} = \|\boldsymbol{\mathcal{X}}_{\text{true}} - \boldsymbol{\mathcal{X}}_{\text{reconstruct}}\|_F / \|\boldsymbol{\mathcal{X}}_{\text{true}}\|_F$. We can see from Fig. 2 that our method consistently outperforms the others in both convergence rate and recovery accuracy in the two cases.

4.2. Video Background and Foreground Separation

Next, we demonstrate the effectiveness of Re-LL1 in performing real video background and foreground separation task on three challenging surveillance video datasets. They are `Highway`, `PETS2006`, and `Pedestrians` from the

[1] Our codes are available at `https://github.com/dangnq2501/Robust-Block-term-Tensor-Decomposition`



(a) Synthetic dataset (b) Office dataset

Fig. 2: Convergence rate of LL1 decomposition methods.

Table 1: Estimation performance of video background and foreground separation methods.

Dataset	`Highway`		`PETS2006`		`Pedestrians`	
Frame size	240 × 320		240 × 320		240 × 360	
No. frame	300		300		300	
Metric	F1	mAP	F1	mAP	F1	mAP
tenRPCA	63.77	47.72	36.65	34.04	67.58	76.48
GRASTA	70.61	**71.15**	42.65	31.48	43.02	17.98
fastRPCA	68.04	61.92	68.11	64.80	77.16	79.61
BRTF	73.38	60.96	**70.08**	**70.40**	77.16	79.62
Re-LL1	**74.58**	62.17	69.30	57.38	**83.83**	**85.14**

CDnet benchmark [21] (see Tab. 1 for their details).[2]

To have a fair comparison, we evaluate our method against four state-of-the-art background/foreground separation methods,[3] including tenRPCA [22], GRASTA [23], fastRPCA [24] and BRTF [25]. Their algorithmic parameters are kept in the default settings. Note that they require a small batch of training video frames to initialize their warm starts.

For Re-LL1, its factors are initialized randomly. Hyperparameters are selected via grid search over a validation subset of the training data. Specifically, the sparsity regularization parameter λ_1 is varied over $\{10^{-3}, 10^{-1}, 1, 10\}$ and the Tikhonov regularization λ_2 over $\{10^{-3}, 10^{-1}\}$ to balance low-rank fidelity with robustness to outliers. The penalty parameter μ and the acceleration step size h are tuned from $\{0.01, 0.1, 1\}$ and $\{0.5, 1.0\}$, respectively. The hyperparameter set is chosen based on the lowest validation reconstruction error and the best F1/mAP scores, and these values are then fixed for all subsequent test datasets.

To measure the performance of separation algorithms, we employ two widely-used metrics: *F1 score* and *mean Average Precision (mAP)*. These metrics evaluate the accuracy and robustness of the detected foreground mask compared to ground truth annotations. The F1 score is the harmonic mean of Precision and Recall, defined as:

$$\mathrm{F1} = 2\frac{\mathrm{Precision} \cdot \mathrm{Recall}}{\mathrm{Precision} + \mathrm{Recall}}, \tag{20}$$

where Precision is the ratio of true positives over predicted positives and Recall is the ratio of true positives over actual positives. While mAP is a threshold-independent metric that evaluates the area under the Precision-Recall curve for each frame, then averages over all frames:

$$\mathrm{mAP} = \frac{1}{N}\sum_{i=1}^{N}\int_0^1 \mathrm{Precision}_i(r)\,dr, \tag{21}$$

where N is the number of video frames and $\mathrm{Precision}_i(r)$ is the precision at recall r for the i-th frame. Their higher values indicate better algorithm performance. We refer the readers to [21, 26–28] for further details.

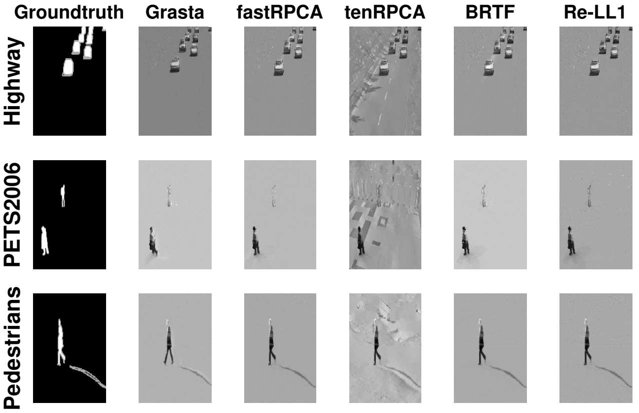


Fig. 3: Video foreground separation results.

Fig. 3 illustrates video foreground separation results while Table 1 reports the F1 score and mAP metrics for all methods across four real videos. Our Re-LL1 method achieves the best performance on the `Highway` (74.58% F1) and the `Pedestrians` (83.83% F1, 85.14% mAP) data. It outperforms tenRPCA by +10.81% in F1 and 14.45% in mAP on the former data, and by 16.25% in F1 and 8.66% in mAP on the latter data. While fastRPCA achieves reasonable separation accuracy on both datasets, its performance is consistently lower than that of Re-LL1. GRASTA achieves the best mAP metric on the `Highway` data, but its F1 score is lower than our method in both datasets. On the `PETS2006` video, Re-LL1 results in 69.30% F1 and 57.38% mAP, closely matching BRTF's top F1 score of 70.08%. It also surpasses fastRPCA, tenRPCA, and GRASTA in F1 score by 1.19%, 32.65%, and 26.65%, respectively.

5. CONCLUSIONS

We have proposed an efficient and effective method for LL1-tensor decomposition, called Re-LL1, which jointly recovers low-rank and identifies sparse components from contaminated tensor data. By integrating Anderson mixing acceleration along with both block-wise and variable-wise updates, Re-LL1 achieves fast convergence. Experiments on synthetic and real video datasets demonstrate that Re-LL1 requires significantly fewer iterations and offers superior reconstruction accuracy compared to standard methods such as LL1-MINF, LL1-NLS, and LL1-BCD. Its effectiveness on real datasets was further demonstrated through video background and foreground separation tasks. Future work will explore adaptive penalty schemes, dynamic rank selection, Bayesian approaches, and extensions to streaming block-term models.

[2] Video data: `http://jacarini.dinf.usherbrooke.ca/`

[3] In the video background and foreground separation task, we omit the comparison of Re-LL1 with LL1-MINF, LL1-NLS, and LL1-BCD, as these methods are not specifically designed for this application.

6. REFERENCES

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

[2] N. D. Sidiropoulos, L. D. Lathauwer, X. Fu, K. Huang *et al.*, “Tensor decomposition for signal processing and machine learning,” *IEEE Trans. Signal Process.*, vol. 65, no. 13, pp. 3551–3582, 2017.

[3] L. T. Thanh, K. Abed-Meraim, N. L. Trung, and A. Hafiane, “A contemporary and comprehensive survey on streaming tensor decomposition,” *IEEE Trans. Knowl. Data Eng.*, vol. 35, no. 11, pp. 10 897–10 921, 2023.

[4] Y. Liu, *Tensors for Data Processing: Theory, Methods, and Applications*. Academic Press, 2021.

[5] H. Fanaee-T and J. Gama, “Tensor-based anomaly detection: An interdisciplinary survey,” *Knowl.-based Syst.*, vol. 98, pp. 130–147, 2016.

[6] L. De Lathauwer, “Decompositions of a higher-order tensor in block terms – Part II: Definitions and uniqueness,” *SIAM J. Matrix Anal. Appl.*, vol. 30, no. 3, pp. 1033–1066, 2008.

[7] L. De Lathauwer and D. Nion, “Decompositions of a higher-order tensor in block terms – Part III: Alternating least squares algorithms,” *SIAM J. Matrix Anal. Appl.*, vol. 30, pp. 1067–1083, 2008.

[8] N. Vervliet, O. Debals, and L. De Lathauwer, “Tensorlab 3.0—numerical optimization strategies for large-scale constrained and coupled matrix/tensor factorization,” in *Proc. Asilomar Conf. Signals, Syst. Comput.*, 2016, pp. 1733–1738.

[9] P. V. Giampouras, A. A. Rontogiannis, and E. Kofidis, “Block-term tensor decomposition model selection and computation: The bayesian way,” *IEEE Trans. Signal Process.*, vol. 70, pp. 1704–1717, 2022.

[10] A. A. Rontogiannis, E. Kofidis, and P. V. Giampouras, “Block-term tensor decomposition: Model selection and computation,” *IEEE J. Sel. Topics Signal Process.*, vol. 15, no. 3, pp. 464–475, 2021.

[11] E. Gujral and E. E. Papalexakis, “OnlineBTD: Streaming algorithms to track the block term decomposition of large tensors,” in *Proc. IEEE Int. Conf. Data Sci. Adv. Anal.*, 2020, pp. 168–177.

[12] L. T. Thanh, K. Abed-Meraim, P. Ravier, and O. Buttelli, “A novel tensor tracking algorithm for block-term decomposition of streaming tensors,” in *Proc. IEEE Stat. Signal Process. Works.*, 2023, pp. 571–575.

[13] A. A. Rontogiannis, E. Kofidis, and P. V. Giampouras, “Online rank-revealing block-term tensor decomposition,” *Signal Process.*, vol. 212, p. 109126, 2023.

[14] E. Kofidis and A. A. Rontogiannis, “Robust rank-revealing block-term decomposition and hyperspectral image denoising,” in *Proc. IEEE Int. Geosci. Remote Sens. Symp.*, 2024, pp. 7639–7643.

[15] F. Xiong, J. Zhou, and Y. Qian, “Hyperspectral restoration via l_0 gradient regularized low-rank tensor factorization,” *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 12, pp. 10 410–10 425, 2019.

[16] L. Sorber, M. Van Barel, and L. De Lathauwer, “Optimization-based algorithms for tensor decompositions: Canonical polyadic decomposition, decomposition in rank-$(\ell_r, \ell_r, 1)$ terms, and a new generalization,” *SIAM J. Opt.*, vol. 23, no. 2, pp. 695–720, 2013.

[17] R. Mazumder, T. Hastie, and R. Tibshirani, “Spectral regularization algorithms for learning large incomplete matrices,” *J. Mach. Learn. Res.*, vol. 11, pp. 2287–2322, 2010.

[18] D. P. Bertsekas, *Constrained Optimization and Lagrange Multiplier Methods*. Academic Press, 2014.

[19] F. Chen, L. Shen, and B. W. Suter, “Computing the proximity operator of the ℓ_p norm with $0 < p < 1$,” *IET Signal Process.*, vol. 10, no. 5, pp. 557–565, 2016.

[20] H. F. Walker and P. Ni, “Anderson acceleration for fixed-point iterations,” *SIAM J. Numer. Anal.*, vol. 49, no. 4, pp. 1715–1735, 2011.

[21] Y. Wang, P.-M. Jodoin, F. Porikli, J. Konrad, and Y. Benezeth, “Cdnet 2014: An expanded change detection benchmark dataset,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recogn. Works.*, 2014, pp. 387–394.

[22] W. Cao, Y. Wang, J. Sun, D. Meng, C. Yang, A. Cichocki, and Z. Xu, “Total variation regularized tensor RPCA for background subtraction from compressive measurements,” *IEEE Trans. Image Process.*, vol. 25, no. 9, pp. 4075–4090, 2016.

[23] J. He, L. Balzano, and J. Huang, “Incremental gradient on the grassmannian for online foreground and background separation in subsampled video,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recogn.*, 2012, pp. 404–411.

[24] P. Zhao, G. Song, and J. Shao, “A fast algorithm for robust principal component analysis via alternating direction method of multipliers,” *IEEE Trans. Image Process.*, vol. 23, no. 6, pp. 2657–2664, 2014.

[25] Q. Zhao, G. Zhou, L. Zhang, A. Cichocki, and S. Amari, “Bayesian robust tensor factorization for incomplete multiway data,” *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 27, no. 4, pp. 736–748, 2016.

[26] C. Li, R. Liu, Z. Quan, P. Hu, and J. Sun, “Foreground separation knowledge distillation for object detection,” *PeerJ Comput. Sci.*, vol. 10, p. e2485, 2024.

[27] R. Margolin, L. Zelnik-Manor, and A. Tal, “How to evaluate foreground maps?” in *Proc. IEEE Conf. Comput. Vis. Pattern Recogn.*, 2014, pp. 248–255.

[28] T. Bouwmans, S. Javed, M. Sultana, and S. K. Jung, “Background modeling and foreground detection for video surveillance: Traditional and recent approaches, challenges and trends,” *Comput. Sci. Rev.*, vol. 22, pp. 1–66, 2014.