Technical University of Denmark

Compressive Online Robust Principal Component Analysis with Multiple Prior Information

Van Luong, Huynh; Deligiannis, Nikos; Seiler, Jürgen; Forchhammer, Søren; Kaup, Andre

Publication date:
2017

Document Version
Publisher's PDF, also known as Version of record

Citation (APA):
Van Luong, H., Deligiannis, N., Seiler, J., Forchhammer, S., & Kaup, A. (2017). Compressive Online Robust Principal Component Analysis with Multiple Prior Information. Paper presented at 5th IEEE Global Conference on Signal and Information Processing, Montreal, Canada.

COMPRESSIVE ONLINE ROBUST PRINCIPAL COMPONENT ANALYSIS WITH MULTIPLE PRIOR INFORMATION

Huynh Van Luong[1], Nikos Deligiannis[2,3], Jürgen Seiler[1], Søren Forchhammer[4], and André Kaup[1]

[1]Multimedia Communications and Signal Processing, Friedrich-Alexander-Universität Erlangen-Nürnberg, 91058 Erlangen, Germany

[2]Department of Electronics and Informatics, Vrije Universiteit Brussel, 1050 Brussels, Belgium

[3]imec, Kapeldreef 75, B-3001 Leuven, Belgium

[4]DTU Fotonik, Technical University of Denmark, 2800 Kgs. Lyngby, Denmark

ABSTRACT

Online Robust Principle Component Analysis (RPCA) arises naturally in time-varying signal decomposition problems such as video foreground-background separation. We propose a compressive online RPCA algorithm that decomposes recursively a sequence of data vectors (e.g., frames) into sparse and low-rank components. Unlike conventional batch RPCA, which processes all the data directly, our method considers a small set of measurements taken per data vector (frame). Moreover, our method incorporates multiple prior information signals, namely previous reconstructed frames, to improve the separation and thereafter, update the prior information for the next frame. Using experiments on synthetic data, we evaluate the separation performance of the proposed algorithm. In addition, we apply the proposed algorithm to online video foreground and background separation from compressive measurements. The results show that the proposed method outperforms the existing methods.

Index Terms— Prior information, robust PCA, n-ℓ_1 minimization, compressive measurements, source separation.

1. INTRODUCTION

Robust Principle Component Analysis (RPCA) [1, 2] decomposes a data matrix $\boldsymbol{M}$ into the sum of unknown sparse $\boldsymbol{S}$ and low-rank $\boldsymbol{L}$ components by solving the Principal Component Pursuit (PCP) [1] problem:

$$\min_{\boldsymbol{L},\boldsymbol{S}} \|\boldsymbol{L}\|_* + \lambda\|\boldsymbol{S}\|_1 \text{ subject to } \boldsymbol{M} = \boldsymbol{L} + \boldsymbol{S}, \tag{1}$$

where $\|\cdot\|_*$ is the matrix nuclear norm (sum of singular values) and $\|\cdot\|_1$ is the ℓ_1-norm. RPCA has found many applications in computer vision, web data analysis, and recommender systems; for example, in video separation, a video sequence is separated into the slowly-changing background (modeled by $\boldsymbol{L}$) and the sparse foreground $\boldsymbol{S}$. However, batch RPCA [1, 2] processes all data samples, e.g., all frames in a video, which involves high computational and memory requirements.

Problem. We consider an online RPCA algorithm that recursively processes a sequence of signals (a.k.a., the column-vectors in $\boldsymbol{M}$) per time instance. Our method recovers the signal from a small set of measurements by leveraging information from a set of previously recovered signals. Formally, at time instance t, we wish to decompose $\boldsymbol{M}_t = \boldsymbol{L}_t + \boldsymbol{S}_t$ into $\boldsymbol{S}_t = [\boldsymbol{x}_1\ \boldsymbol{x}_1\ ...\ \boldsymbol{x}_t]$ and $\boldsymbol{L}_t = [\boldsymbol{v}_1\ \boldsymbol{v}_2\ ...\ \boldsymbol{v}_t]$, where $[\cdot]$ denotes a matrix and $\boldsymbol{x}_t, \boldsymbol{v}_t \in \mathbb{R}^n$ are column-vectors in $\boldsymbol{S}_t$ and $\boldsymbol{L}_t$, respectively. We assume that $\boldsymbol{L}_{t-1}$ and $\boldsymbol{S}_{t-1}$ have been recovered at time instance $t-1$ and that at time instance t we have access to compressive measurements of the vector $\boldsymbol{x}_t + \boldsymbol{v}_t$, that is, we observe $\boldsymbol{y}_t = \boldsymbol{\Phi}(\boldsymbol{x}_t + \boldsymbol{v}_t)$, where $\boldsymbol{\Phi} \in \mathbb{R}^{m\times n}$ $(m<n)$ is a random projection [3]. The recovery problem at time instance t is thus written as

$$\min_{\boldsymbol{x}_t,\boldsymbol{v}_t} \|[\boldsymbol{L}_{t-1}\ \boldsymbol{v}_t]\|_* + \lambda\|\boldsymbol{x}_t\|_1 \text{ subject to } \boldsymbol{y}_t = \boldsymbol{\Phi}(\boldsymbol{x}_t + \boldsymbol{v}_t), \tag{2}$$

where $\boldsymbol{L}_{t-1}=[\boldsymbol{v}_1\ \boldsymbol{v}_2\ ...\ \boldsymbol{v}_{t-1}]$, $\boldsymbol{S}_{t-1}=[\boldsymbol{x}_1\ \boldsymbol{x}_1\ ...\ \boldsymbol{x}_{t-1}]$, $\boldsymbol{\Phi}$ are given.

Related Work. Incremental PCP [4] processes each column-vector in $\boldsymbol{M}$ at a time, assuming access to the complete data (e.g., full frames) rather than compressive measurements. Compressive PCP [5], on the other hand, is the counterpart of batch RPCA that operates on compressive measurements. Other related studies [6–9] addressed the problem of online estimation of low-dimensional subspaces from randomly subsampled data. In [10], an algorithm was proposed to recover the sparse component $\boldsymbol{x}_t$ in (2) by solving the problem $\min_{\boldsymbol{x}_t} \|\boldsymbol{x}_t\|_1$ subject to $\boldsymbol{y}_t = \boldsymbol{\Phi}_t(\boldsymbol{A}\boldsymbol{x}_t + \boldsymbol{B}\boldsymbol{v}_t)$, where $\boldsymbol{\Phi}_t \in \mathbb{R}^{m\times m}$ and $\boldsymbol{A}, \boldsymbol{B} \in \mathbb{R}^{m\times n}$. However, the low-rank component $\boldsymbol{v}_t$ in (2) was not recovered per time instance. Alternatively, the studies in [11], [12] assumed the low-rank vector $\boldsymbol{v}_t$ not-varying and proposed a method to estimate the number of compressive measurements required to recover $\boldsymbol{x}_t$ per time instance.

The problem of reconstructing a sequence of time-varying sparse signals using prior information is also playing an important role in the context of online RPCA [10, 13, 14]. There were several studies on sparse signal recovery from low-dimensional measurements that proposed to leverage some form of prior information [10, 11, 13, 15]. The study in [15] provided a comprehensive overview of the domain, reviewing a class of recursive algorithms. The studies in [10, 13] used modified-CS [16] to leverage prior knowledge under the condition of slowly varying support and signal values. Recently, the study in [17] presented an online compressive RPCA method that is supported by performance guarantees. However, this method as well as the methods in [6, 7, 9] do not explore the correlation across adjacent sparse components from multiple previously recovered frames.

Contributions. We propose a *compressive online RPCA with multiple prior information* (CORPCA) algorithm, which leverages information from previously recovered sparse components by utilizing RAMSIA—our previously-proposed algorithm for sparse signal recovery with multiple prior information signals [18]—and exploits the slowly-changing characteristics of low-rank components via an incremental SVD [19] method. We solve the compressive decomposition problem in (2) in an online manner by minimizing, *(i)* an n-ℓ_1-norm cost function [18] for the sparse part; and *(ii)* the rank of a matrix for the low-rank part.

2. PROBLEM FORMULATION AND ALGORITHM

2.1. Problem Formulation

The proposed CORPCA algorithm is based on RAMSIA [18], our algorithm that uses n-ℓ_1 minimization with adaptive weights to recover a sparse signal $\boldsymbol{x}$ from low-dimensional random measurements $\boldsymbol{y} = \boldsymbol{\Phi}\boldsymbol{x}$ with the aid of multiple side information signals $\boldsymbol{z}_j$, $j \in$

$\{0, 1, \ldots, J\}$, with $\boldsymbol{z}_0 = \mathbf{0}$. The objective function of RAMSIA [18] is given by

$$\min_{\boldsymbol{x}}\Big\{H(\boldsymbol{x})=\frac{1}{2}\|\boldsymbol{\Phi}\boldsymbol{x}-\boldsymbol{y}\|_2^2+\lambda\sum_{j=0}^{J}\beta_j\|\mathbf{W}_j(\boldsymbol{x}-\boldsymbol{z}_j)\|_1\Big\}, \quad (3)$$

where $\lambda > 0$ and $\beta_j > 0$ are weights across the side information signals, and $\mathbf{W}_j$ is a diagonal matrix with weights for each element in the side information signal $\boldsymbol{z}_j$; namely, $\mathbf{W}_j = \mathrm{diag}(w_{j1}, w_{j2}, ..., w_{jn})$ with $w_{ji} > 0$ being the weight for the i-th element in the $\boldsymbol{z}_j$ vector.

The proposed CORPCA method aims at processing one data vector per time instance by leveraging prior information for both its sparse and low-rank components. At time instance t, we observe $\boldsymbol{y}_t = \boldsymbol{\Phi}(\boldsymbol{x}_t + \boldsymbol{v}_t)$ with $\boldsymbol{y}_t \in \mathbb{R}^m$. Let $\boldsymbol{Z}_{t-1} := \{\boldsymbol{z}_1, ..., \boldsymbol{z}_J\}$, a set of $\boldsymbol{z}_j \in \mathbb{R}^n$, and $\boldsymbol{B}_{t-1} \in \mathbb{R}^{n\times d}$ denote prior information for $\boldsymbol{x}_t$ and $\boldsymbol{v}_t$, respectively. As discussed in Sec. 2.2, we form the prior information $\boldsymbol{Z}_{t-1}$ and $\boldsymbol{B}_{t-1}$ using the already reconstructed set of vectors $\{\hat{\boldsymbol{x}}_1, ..., \hat{\boldsymbol{x}}_{t-1}\}$ and $\{\hat{\boldsymbol{v}}_1, ..., \hat{\boldsymbol{v}}_{t-1}\}$.

To solve the problem in (2), we formulate the objective function of CORPCA as

$$\min_{\boldsymbol{x}_t,\boldsymbol{v}_t}\Big\{H(\boldsymbol{x}_t,\boldsymbol{v}_t|\boldsymbol{y}_t,\boldsymbol{Z}_{t-1},\boldsymbol{B}_{t-1})=\frac{1}{2}\|\boldsymbol{\Phi}(\boldsymbol{x}_t+\boldsymbol{v}_t)-\boldsymbol{y}_t\|_2^2$$
$$+\lambda\mu\sum_{j=0}^{J}\beta_j\|\mathbf{W}_j(\boldsymbol{x}_t-\boldsymbol{z}_j)\|_1+\mu\Big\|[\boldsymbol{B}_{t-1}\ \boldsymbol{v}_t]\Big\|_*\Big\}, \quad (4)$$

where $\mu > 0$. It can be seen that when $\boldsymbol{v}_t$ is static (not changing), Problem (4) would become Problem (3). Furthermore, when $\boldsymbol{x}_t$ and $\boldsymbol{v}_t$ are batch variables and we do not take the prior information, $\boldsymbol{Z}_{t-1}$ and $\boldsymbol{B}_{t-1}$, and the projection $\boldsymbol{\Phi}$ into account, Problem (4) becomes Problem (1).

2.2. CORPCA Algorithm

The proposed algorithm operates in two steps: Firstly, we solve Problem (4) given that $\boldsymbol{Z}_{t-1}$ and $\boldsymbol{B}_{t-1}$ are known (they are obtained from the time instance or recursion). Thereafter, we update $\boldsymbol{Z}_t$ and $\boldsymbol{B}_t$, which are used in the following time instance.

Solution of Problem (4). Let us denote $f(\boldsymbol{v}_t, \boldsymbol{x}_t) = (1/2)\|\boldsymbol{\Phi}(\boldsymbol{x}_t+\boldsymbol{v}_t)-\boldsymbol{y}_t\|_2^2$, $g(\boldsymbol{x}_t) = \lambda\sum_{j=0}^{J}\beta_j\|\mathbf{W}_j(\boldsymbol{x}_t-\boldsymbol{z}_j)\|_1$, and $h(\boldsymbol{v}_t) = \|[\boldsymbol{B}_{t-1}\ \boldsymbol{v}_t]\|_*$. The solution of (4) is obtained by the proposed CORPCA algorithm in Algorithm 1 (the code is online [20]) using proximal gradient methods [2, 21]. Specifically, as shown in Lines 3-9 in Algorithm 1, we iteratively compute $\boldsymbol{x}_t^{(k+1)}$ and $\boldsymbol{v}_t^{(k+1)}$ at iteration $k+1$ via the soft thresholding operator [21] for $\boldsymbol{x}_t$ and the single value thresholding operator [22] for $\boldsymbol{v}_t$:

$$\boldsymbol{v}_t^{(k+1)}=\arg\min_{\boldsymbol{v}_t}\Big\{\mu h(\boldsymbol{v}_t)+\Big\|\boldsymbol{v}_t-\Big(\boldsymbol{v}_t^{(k)}-\frac{1}{2}\nabla_{\boldsymbol{v}_t}f(\boldsymbol{v}_t^{(k)},\boldsymbol{x}_t^{(k)})\Big)\Big\|_2^2\Big\}, \quad (5)$$

$$\boldsymbol{x}_t^{(k+1)}=\arg\min_{\boldsymbol{x}_t}\Big\{\mu g(\boldsymbol{x}_t)+\Big\|\boldsymbol{x}_t-\Big(\boldsymbol{x}_t^{(k)}-\frac{1}{2}\nabla_{\boldsymbol{x}_t}f(\boldsymbol{v}_t^{(k)},\boldsymbol{x}_t^{(k)})\Big)\Big\|_2^2\Big\}. \quad (6)$$

The proximal operator $\boldsymbol{\Gamma}_{\tau g_1}(\cdot)$ in Line 7 of Algorithm 1 is defined as

$$\boldsymbol{\Gamma}_{\tau g_1}(\boldsymbol{X})=\arg\min_{\boldsymbol{V}}\Big\{\tau g_1(\boldsymbol{V})+\frac{1}{2}||\boldsymbol{V}-\boldsymbol{X}||_2^2\Big\}, \quad (7)$$

where $g_1(\cdot) = \|\cdot\|_1$. The weights $\mathbf{W}_j$ and β_j are updated per iteration of the algorithm (see Lines 10-11). As suggested in [2], the convergence of Algorithm 1 in Line 2 is determined by evaluating the criterion $\|\partial H(\boldsymbol{x}_t,\boldsymbol{v}_t)|_{\boldsymbol{x}_t^{(k+1)},\boldsymbol{v}_t^{(k+1)}}\|_2^2 < 2 * 10^{-7}\|(\boldsymbol{x}_t^{(k+1)},\boldsymbol{v}_t^{(k+1)})\|_2^2$. Finally, we update the prior information for the next instance, $\boldsymbol{Z}_t$ and $\boldsymbol{B}_t$, in Lines 15-16.

Prior Information Update. The update of $\boldsymbol{Z}_t$ and $\boldsymbol{B}_t$ is carried out after each time instance (see Lines 15-16, Algorithm 1). Due to the correlation between subsequent signals (e.g., frames),

Algorithm 1: The proposed CORPCA algorithm.

Input: $\boldsymbol{y}_t,\ \boldsymbol{\Phi},\ \boldsymbol{Z}_{t-1},\ \boldsymbol{B}_{t-1}$;
Output: $\widehat{\boldsymbol{x}}_t,\ \widehat{\boldsymbol{v}}_t,\ \boldsymbol{Z}_t,\ \boldsymbol{B}_t$;
`// Initialize variables and parameters.`
1 $\boldsymbol{x}_t^{(-1)}=\boldsymbol{x}_t^{(0)}=\mathbf{0}$; $\boldsymbol{v}_t^{(-1)}=\boldsymbol{v}_t^{(0)}=\mathbf{0}$; $\xi_{-1}=\xi_0=1$; $\mu_0=0$; $\bar{\mu}>0$; $\lambda>0$; $0<\epsilon<1$; $k=0$; $g_1(\cdot)=\|\cdot\|_1$;
2 **while** *not converged* **do**
`// Solve Problem (4).`
3 $\widetilde{\boldsymbol{v}_t}^{(k)}=\boldsymbol{v}_t^{(k)}+\frac{\xi_{k-1}-1}{\xi_k}(\boldsymbol{v}_t^{(k)}-\boldsymbol{v}_t^{(k-1)})$;
4 $\widetilde{\boldsymbol{x}_t}^{(k)}=\boldsymbol{x}_t^{(k)}+\frac{\xi_{k-1}-1}{\xi_k}(\boldsymbol{x}_t^{(k)}-\boldsymbol{x}_t^{(k-1)})$;
5 $\nabla_{\boldsymbol{v}_t}f(\widetilde{\boldsymbol{v}_t}^{(k)},\widetilde{\boldsymbol{x}_t}^{(k)})=\nabla_{\boldsymbol{x}_t}f(\widetilde{\boldsymbol{v}_t}^{(k)},\widetilde{\boldsymbol{x}_t}^{(k)})=\boldsymbol{\Phi}^{\mathrm{T}}\Big(\boldsymbol{\Phi}(\widetilde{\boldsymbol{v}_t}^{(k)}+\widetilde{\boldsymbol{x}_t}^{(k)})-\boldsymbol{y}_t\Big)$;
6 $(\boldsymbol{U}_t,\boldsymbol{\Sigma}_t,\boldsymbol{V}_t)=\mathrm{incSVD}\Big(\Big[\boldsymbol{B}_{t-1}\ \Big(\widetilde{\boldsymbol{v}_t}^{(k)}-\frac{1}{2}\nabla_{\boldsymbol{v}_t}f(\widetilde{\boldsymbol{v}_t}^{(k)},\widetilde{\boldsymbol{x}_t}^{(k)})\Big)\Big]\Big)$;
7 $\boldsymbol{\Theta}_t=\boldsymbol{U}_t\boldsymbol{\Gamma}_{\frac{\mu_k}{2}g_1}(\boldsymbol{\Sigma}_t)\boldsymbol{V}_t^T$;
8 $\boldsymbol{v}_t^{(k+1)}=\boldsymbol{\Theta}_t(:,\mathrm{end})$;
9 $\boldsymbol{x}_t^{(k+1)}=\Gamma_{\frac{\mu_k}{2}g}\Big(\widetilde{\boldsymbol{x}_t}^{(k)}-\frac{1}{2}\nabla_{\boldsymbol{x}_t}f(\widetilde{\boldsymbol{v}_t}^{(k)},\widetilde{\boldsymbol{x}_t}^{(k)})\Big)$; where $\Gamma_{\frac{\mu_k}{2}g}(\cdot)$ is given as in RAMSIA [18];
`// Compute the updated weights [18].`
10 $w_{ji}=\dfrac{n(|x_{ti}^{(k+1)}-z_{ji}|+\epsilon)^{-1}}{\sum_{l=1}^{n}(|x_{tl}^{(k+1)}-z_{jl}|+\epsilon)^{-1}}$;
11 $\beta_j=\dfrac{\Big(||\mathbf{W}_j(\boldsymbol{x}_t^{(k+1)}-\boldsymbol{z}_j)||_1+\epsilon\Big)^{-1}}{\sum_{l=0}^{J}\Big(||\mathbf{W}_l(\boldsymbol{x}_t^{(k+1)}-\boldsymbol{z}_l)||_1+\epsilon\Big)^{-1}}$;
12 $\xi_{k+1}=(1+\sqrt{1+4\xi_k^2})/2$; $\mu_{k+1}=\max(\epsilon\mu_k,\bar{\mu})$;
13 $k=k+1$;
14 **end**
`// Update prior information.`
15 $\boldsymbol{Z}_t:=\{\boldsymbol{z}_j=\boldsymbol{x}_{t-J+j}^{(k+1)}\}_{j=1}^{J}$;
16 $\boldsymbol{B}_t=\boldsymbol{U}_t(:,1:d)\boldsymbol{\Gamma}_{\frac{\mu_k}{2}g_1}(\boldsymbol{\Sigma}_t)(1:d,1:d)\boldsymbol{V}_t(:,1:d)^{\mathrm{T}}$;
17 **return** $\widehat{\boldsymbol{x}}_t=\boldsymbol{x}_t^{(k+1)},\ \widehat{\boldsymbol{v}}_t=\boldsymbol{v}_t^{(k+1)},\ \boldsymbol{Z}_t,\ \boldsymbol{B}_t$;

we update the prior information $\boldsymbol{Z}_t$ by using the J latest recovered sparse components, that is, $\boldsymbol{Z}_t := \{\boldsymbol{z}_j = \boldsymbol{x}_{t-J+j}\}_{j=1}^{J}$. For $\boldsymbol{B}_t \in \mathbb{R}^{n\times d}$, we consider an adaptive update, which operates on a fixed or constant number d of the columns of $\boldsymbol{B}_t$. To this end, we use the incremental singular decomposition SVD [19] method (incSVD$(\cdot)$ in Line 6, Algorithm 1). It is worth noting that the update $\boldsymbol{B}_t = \boldsymbol{U}_t\boldsymbol{\Gamma}_{\frac{\mu_k}{2}g_1}(\boldsymbol{\Sigma}_t)\boldsymbol{V}_t^{\mathrm{T}}$, causes the dimension of $\boldsymbol{B}_t$ to increase as $\boldsymbol{B}_t \in \mathbb{R}^{n\times(d+1)}$ after each instance. However, in order to maintain a reasonable number of d, we take $\boldsymbol{B}_t = \boldsymbol{U}_t(:,1:d)\boldsymbol{\Gamma}_{\frac{\mu_k}{2}g_1}(\boldsymbol{\Sigma}_t)(1:d,1:d)\boldsymbol{V}_t(:,1:d)^{\mathrm{T}}$. The computational cost of incSVD$(\cdot)$ is lower than conventional SVD [4, 19] since we only compute the full SVD of the middle matrix with size $(d+1)\times(d+1)$, where $d \ll n$, instead of $n\times(d+1)$.

The computation of incSVD$(\cdot)$ is presented in the following: The goal is to compute incSVD$[\boldsymbol{B}_{t-1}\ \boldsymbol{v}_t]$, i.e., $[\boldsymbol{B}_{t-1}\ \boldsymbol{v}_t] = \boldsymbol{U}_t\boldsymbol{\Sigma}_t\boldsymbol{V}_t^{\mathrm{T}}$. By taking the SVD of $\boldsymbol{B}_{t-1} \in \mathbb{R}^{n\times d}$ we obtain $\boldsymbol{B}_{t-1} = \boldsymbol{U}_{t-1}\boldsymbol{\Sigma}_{t-1}\boldsymbol{V}_{t-1}^{\mathrm{T}}$. Therefore, we can derive $(\boldsymbol{U}_t, \boldsymbol{\Sigma}_t, \boldsymbol{V}_t)$ via $(\boldsymbol{U}_{t-1}, \boldsymbol{\Sigma}_{t-1}, \boldsymbol{V}_{t-1})$ and $\boldsymbol{v}_t$. We write the matrix $[\boldsymbol{B}_{t-1}\ \boldsymbol{v}_t]$ as

$$[\boldsymbol{B}_{t-1}\ \boldsymbol{v}_t]=\Big[\boldsymbol{U}_{t-1}\ \frac{\boldsymbol{\delta}_t}{\|\boldsymbol{\delta}_t\|_2}\Big]\cdot\begin{bmatrix}\boldsymbol{\Sigma}_{t-1} & \boldsymbol{e}_t\\ \mathbf{0}^{\mathrm{T}} & \|\boldsymbol{\delta}_t\|_2\end{bmatrix}\cdot\begin{bmatrix}\boldsymbol{V}_{t-1}^{\mathrm{T}} & \mathbf{0}\\ \mathbf{0}^{\mathrm{T}} & 1\end{bmatrix}, \quad (8)$$

where $\boldsymbol{e}_t = \boldsymbol{U}_{t-1}^{\mathrm{T}}\boldsymbol{v}_t$ and $\boldsymbol{\delta}_t = \boldsymbol{v}_t - \boldsymbol{U}_{t-1}\boldsymbol{e}_t$. By taking the

Table 1. Summary of recovery characteristics

		CORPCA	RPCA [1]	GRASTA [6]	ReProCS [10]
Online		✓		✓	✓
Full data		✓	✓	✓	✓
Compressed	Foreground	✓			✓
	Background	✓		✓	

SVD of the matrix in between the right side of (8), we obtain $\begin{bmatrix} \boldsymbol{\Sigma}_{t-1} & \boldsymbol{e}_t \\ \mathbf{0}^{\mathrm{T}} & \|\boldsymbol{\delta}_t\|_2 \end{bmatrix} = \widetilde{\boldsymbol{U}}\widetilde{\boldsymbol{\Sigma}}\widetilde{\boldsymbol{V}}^{\mathrm{T}}$. Eventually, we obtain $\boldsymbol{U}_t = \left[\boldsymbol{U}_{t-1} \ \frac{\boldsymbol{\delta}_t}{\|\boldsymbol{\delta}_t\|_2}\right] \cdot \widetilde{\boldsymbol{U}}$, $\boldsymbol{\Sigma}_t = \widetilde{\boldsymbol{\Sigma}}$, and $\boldsymbol{V}_t = \begin{bmatrix} \boldsymbol{V}_{t-1}^{\mathrm{T}} & \mathbf{0} \\ \mathbf{0}^{\mathrm{T}} & 1 \end{bmatrix} \cdot \widetilde{\boldsymbol{V}}$.

3. EXPERIMENTAL RESULTS

We evaluate the performance of our Algorithm 1 employing the proposed method, the existing ℓ_1 minimization [21], and the existing ℓ_1-ℓ_1 [23, 24] minimization methods, denoted as CORPCA-n-ℓ_1, CORPCA-ℓ_1, and CORPCA-ℓ_1-ℓ_1, respectively. We also compare CORPCA against RPCA [1], GRASTA [6], and ReProCS [10], the characteristics of which are summarized in Table 1. RPCA [1] is a batch-based method assuming full access to the data, while GRASTA [6] and ReProCS [10] are online methods that can recover either the low-rank component (GRASTA) or the sparse component (ReProCS) from compressive measurements.

3.1. Experiments with Synthetic Data

We generate our data as follows. We generate the low-rank component as $\boldsymbol{L} = \boldsymbol{U}\boldsymbol{V}^{\mathrm{T}}$, where $\boldsymbol{U} \in \mathbb{R}^{n \times r}$ and $\boldsymbol{V} \in \mathbb{R}^{(d+q) \times r}$ are random matrices whose entries are drawn from the standard normal distribution. We set $n = 500$, $r = 5$ (rank of $\boldsymbol{L}$), and $d = 100$ the number of vectors for training and $q = 100$ the number of testing vectors. This yields $\boldsymbol{L} = [\boldsymbol{v}_1 \ ... \ \boldsymbol{v}_{d+q}]$. We generate $\boldsymbol{S} = [\boldsymbol{x}_1 \ ... \ \boldsymbol{x}_{d+q}]$, where at time instance $t = 1$, $\boldsymbol{x}_1 \in \mathbb{R}^n$ is generated from the standard normal distribution with s_0 nonzero elements, denoted by $\|\boldsymbol{x}_1\|_0 = s_0$. Our purpose is to consider a sequence of correlated sparse vectors $\boldsymbol{x}_t$ with $t > 1$. Therefore, we generate $\boldsymbol{x}_t$ satisfying $\|\boldsymbol{x}_t - \boldsymbol{x}_{t-1}\|_0 = s_0/2$. This could lead to $\|\boldsymbol{x}_t\|_0 > s_0$. To avoid a large increase of $\|\boldsymbol{x}_t\|_0$, we constrain $\|\boldsymbol{x}_t\|_0 \in [s_0, s_0 + 15]$, whenever $\|\boldsymbol{x}_t\|_0 > s_0 + 15$, $\boldsymbol{x}_t$ is randomly reset to $\|\boldsymbol{x}_t\|_0 = s_0$ by setting $\|\boldsymbol{x}_t\|_0 - s_0$ positions that are randomly selected to zero. Here, we test our algorithms for $s_0 = 10$ to 90.

The prior information is initialized as follows. To address real scenarios, where we do not know the sparse and low-rank components, we use the batch-based RPCA [1] to separate the training set $\boldsymbol{M}_0 = [\boldsymbol{x}_1 + \boldsymbol{v}_1 \ ... \ \boldsymbol{x}_d + \boldsymbol{v}_d]$ so as to obtain $\boldsymbol{B}_0 = [\boldsymbol{v}_1 \ ... \ \boldsymbol{v}_d]$. In this experiment, we use three ($J = 3$) sparse components as prior information and we set $\boldsymbol{Z}_0 := \{\mathbf{0}, \mathbf{0}, \mathbf{0}\}$. We run CORPCA (Sec. 2.2) on the test set $\boldsymbol{M} = [\boldsymbol{x}_{d+1} + \boldsymbol{v}_{d+1} \ ... \ \boldsymbol{x}_{d+q} + \boldsymbol{v}_{d+q}]$.

We assess the accuracy of recovering $\widehat{\boldsymbol{x}}_t, \widehat{\boldsymbol{v}}_t$ versus $\boldsymbol{x}_t, \boldsymbol{v}_t$ in terms of the success probability, denoted as $\Pr(\text{success})$, versus the number of measurements m, which is the dimensionality of vector $\boldsymbol{y}_t$. For instance, for $\widehat{\boldsymbol{x}}_t$ given a fixed m, $\Pr(\text{success})$ is the number of times, in which the source $\boldsymbol{x}_t$ is recovered as $\widehat{\boldsymbol{x}}_t$ with an error $\|\widehat{\boldsymbol{x}}_t - \boldsymbol{x}_t\|_2 / \|\boldsymbol{x}_t\|_2 \leq 10^{-2}$, divided by the total 50 Monte Carlo simulations and where we have set $\epsilon = 0.8$, $\lambda = 1/\sqrt{n}$.

The results in Fig. 1 demonstrate the efficiency of the proposed CORPCA employing n-ℓ_1 minimization. At specific sparsity levels, we can recover the 500-dimensional data from measurements of much lower dimensions [$m = 150$ to 300, see the white areas in Fig. 1(a)]. It is also clear that the ℓ_1 and ℓ_1-ℓ_1 minimization methods [see Figs. 1(b), 1(c)] lead to a higher number of measurements, thereby illustrating the benefit of incorporating multiple side information into the problem. As mentioned in Table 1, ReProCS and GRASTA can only recover the foreground and background from compressive measurements. Fig. 1(d) shows that the efficiency of ReProCS is worse than that of CORPCA-n-ℓ_1 [see Fig. 1(a)]. Furthermore, Fig. 1(e) shows that GRASTA delivers poor background recovery. For further results on the measurement bounds of CORPCA, we refer to our work in [25].

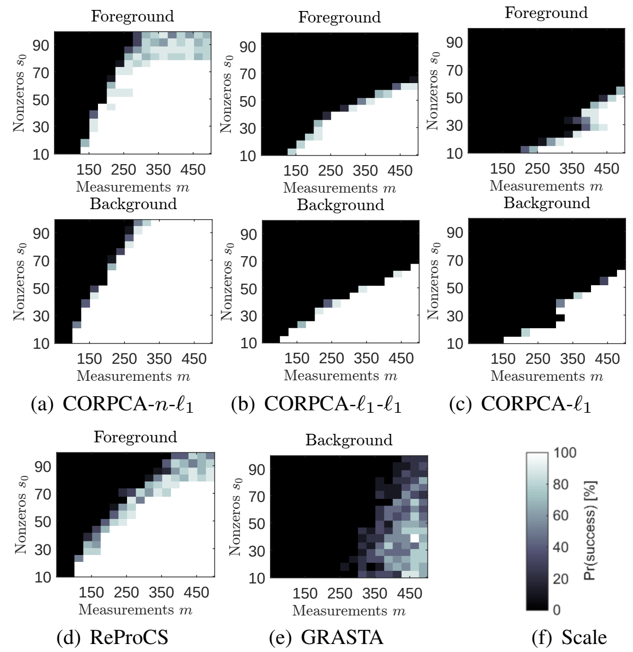


Fig. 1. Average success probabilities for CORPCA (for $\boldsymbol{x}_t, \boldsymbol{v}_t$), ReProCS (for $\boldsymbol{x}_t$), and GRASTA (for $\boldsymbol{v}_t$). The scale (f) is proportional to $\Pr(\text{success})[\%]$ from black to white.

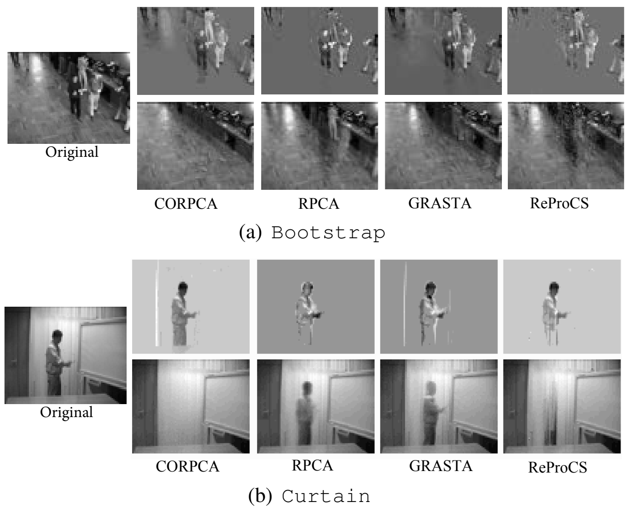
3.2. Compressive Video Foreground-Background Separation

We assess our CORPCA method in the application of compressive video separation and compare it against the existing methods. We run all methods listed in Table 1 on typical test video content [26]. In this experiment, we use $d = 100$ frames as training vectors for the proposed CORPCA as well as for GRASTA [6] and ReProCS [10], and three latest previous foregrounds as prior sparse information.

3.2.1. *Visual Evaluation*

We consider two videos [26], `Bootstrap` (60×80 pixels) and `Curtain` (64×80 pixels) [c.f., Fig. 2], having a static and a dynamic background, respectively. We first consider background-foreground video separation with full access to the video data (the data set $\boldsymbol{M}$); the visual results of the various methods are illustrated in Fig. 2. It is evident that, for both the video sequences, CORPCA delivers superior visual results than the other methods, which suffer from less-details in the foreground and noisy background images.

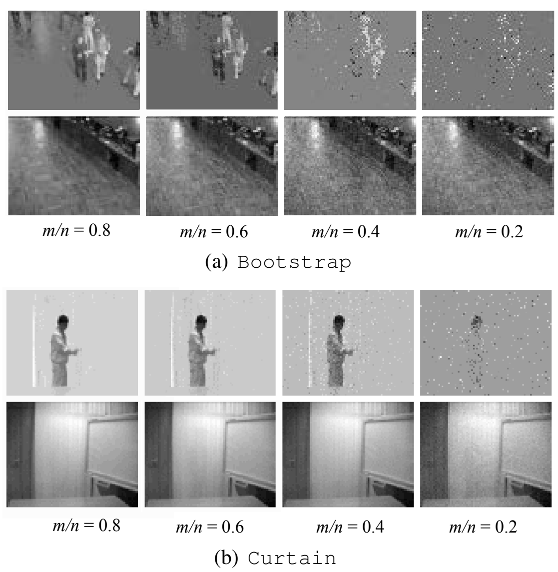
Fig. 3 presents the results of CORPCA under various rates on the number of measurements m over the dimension n of the data (the size of the vectorized frame). The results show that we can recover the foreground and background even by accessing a small number of measurements; for instance, we can obtain good-quality reconstructions with only $m/n = 0.6$ and $m/n = 0.4$ for `Bootstrap` [see



(a) `Bootstrap`

(b) `Curtain`

Fig. 2. Background and foreground separation for the different separation methods with full data access `Bootstrap#2213` and `Curtain#2866`.



(a) `Bootstrap`

(b) `Curtain`

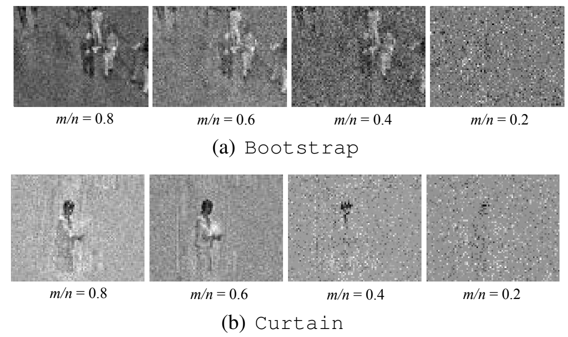
Fig. 3. Compressive background and foreground separation of CORPCA with different measurement rates m/n.

Fig. 3(a)] and `Curtain` [see Fig. 3(b)], respectively. `Bootstrap` requires more measurements than `Curtain` due to the more complex foreground information. For comparison, we illustrate the visual results obtained with ReProCS—which, however, can only recover the foreground using compressive measurements—in Fig. 4. It is clear that the reconstructed foreground images have a poorer visual quality compared to CORPCA even at a high rate $m/n = 0.8$[1].

3.2.2. *Quantitative Results*

We evaluate quantitatively the separation performance via the *receiver operating curve* (ROC) metric [27]. The metrics *True positives* and *False positives* are defined as in [27]. Fig. 5 illustrates the ROC results when assuming full data access, i.e., $m/n = 1$, of

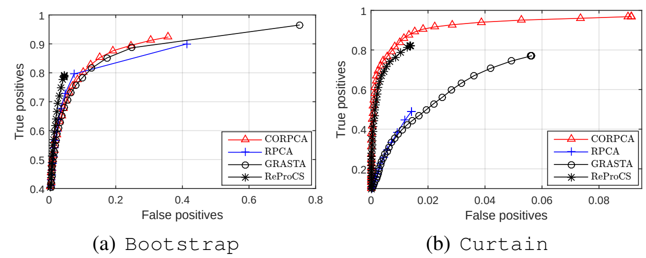
[1]The original test videos and the reconstructed separated sequences are available online [20].



(a) `Bootstrap`

(b) `Curtain`

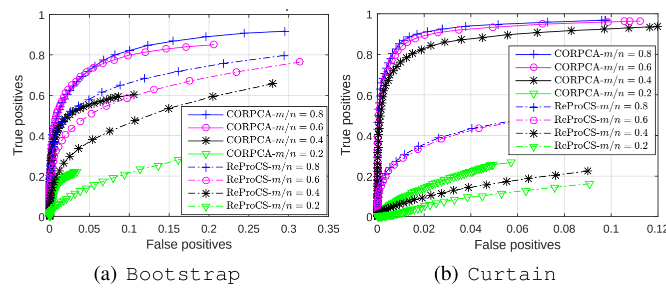
Fig. 4. Compressive foreground separation of ReProCS with different measurement rates m/n.



(a) `Bootstrap`

(b) `Curtain`

Fig. 5. ROC for the different separation methods with full data.



(a) `Bootstrap`

(b) `Curtain`

Fig. 6. ROC for CORPCA and ReProCS with compressive measurements.

CORPCA, RPCA, GRASTA, and ReProCS. The results show that CORPCA delivers higher performance than the other methods, especially for the `Curtain` video sequence [c.f., Fig. 5(b)]. Furthermore, we compare the foreground recovery performance of CORPCA against ReProCS for different compressive measurement rates: $m/n = \{0.8; 0.6; 0.4; 0.2\}$. The ROC results in Fig. 6 show that CORPCA achieves a relatively high performance with small number of measurements: for `Bootstrap` until $m/n = 0.6$ [see Fig. 6(a)] and for `Curtain` until $m/n = 0.4$ [see Fig. 6(b)]. The ROC results for ReProCS are quickly degrading even with a high compressive measurement rate $m/n = 0.8$.

4. CONCLUSION

This paper proposed a compressive online robust PCA algorithm (CORPCA) that can process a data vector per time instance using compressive measurements. CORPCA efficiently incorporates multiple prior information based on the n-ℓ_1 minimization problem. We have tested our method on synthetic data as well as in the compressive video separation application using video data. The results revealed the advantage of incorporating prior information by employing n-ℓ_1 minimization and demonstrated the superior performance improvement offered by CORPCA compared to existing methods.

5. REFERENCES

[1] E. J. Candès, X. Li, Y. Ma, and J. Wright, "Robust principal component analysis?" *J. ACM*, vol. 58, no. 3, pp. 11:1–11:37, Jun. 2011.

[2] J. Wright, A. Ganesh, S. Rao, Y. Peng, and Y. Ma, "Robust principal component analysis: Exact recovery of corrupted low-rank matrices via convex optimization," in *Advances in Neural Information Processing Systems 2*, 2009.

[3] E. Candès and T. Tao, "Near-optimal signal recovery from random projections: Universal encoding strategies?" *IEEE Trans. Inf. Theory*, vol. 52, no. 12, pp. 5406–5425, Apr. 2006.

[4] P. Rodriguez and B. Wohlberg, "Incremental principal component pursuit for video background modeling," *Journal of Mathematical Imaging and Vision*, vol. 55, no. 1, pp. 1–18, 2016.

[5] J. Wright, A. Ganesh, K. Min, and Y. Ma, "Compressive principal component pursuit," *Information and Inference*, vol. 2, no. 1, pp. 32–68, 2013.

[6] J. He, L. Balzano, and A. Szlam, "Incremental gradient on the grassmannian for online foreground and background separation in subsampled video," in *IEEE Conference on Computer Vision and Pattern Recognition*, June 2012.

[7] J. Xu, V. K. Ithapu, L. Mukherjee, J. M. Rehg, and V. Singh, "Gosus: Grassmannian online subspace updates with structured-sparsity," in *IEEE International Conference on Computer Vision*, Dec 2013.

[8] J. Feng, H. Xu, and S. Yan, "Online robust pca via stochastic optimization," in *Advances in Neural Information Processing Systems 26*, 2013.

[9] H. Mansour and X. Jiang, "A robust online subspace estimation and tracking algorithm," in *IEEE Int. Conf. on Acoustics, Speech and Signal Processing*, April 2015.

[10] H. Guo, C. Qiu, and N. Vaswani, "An online algorithm for separating sparse and low-dimensional signal sequences from their sum," *IEEE Trans. Signal Process.*, vol. 62, no. 16, pp. 4284–4297, 2014.

[11] J. F. Mota, N. Deligiannis, A. C. Sankaranarayanan, V. Cevher, and M. R. Rodrigues, "Adaptive-rate reconstruction of time-varying signals with application in compressive foreground extraction," *IEEE Trans. Signal Process.*, vol. 64, no. 14, pp. 3651–3666, 2016.

[12] G. Warnell, S. Bhattacharya, R. Chellappa, and T. Basar, "Adaptive-rate compressive sensing using side information," *IEEE Trans. Image Process.*, vol. 24, no. 11, pp. 3846–3857, 2015.

[13] C. Qiu, N. Vaswani, B. Lois, and L. Hogben, "Recursive robust PCA or recursive sparse recovery in large but structured noise," *IEEE Trans. Inf. Theory*, vol. 60, no. 8, pp. 5007–5039, 2014.

[14] J. Zhan and N. Vaswani, "Robust pca with partial subspace knowledge," in *IEEE Int. Symposium on Information Theory*, 2014.

[15] N. Vaswani and J. Zhan, "Recursive recovery of sparse signal sequences from compressive measurements: A review," *IEEE Trans. Signal Process.*, vol. 64, no. 13, pp. 3523–3549, 2016.

[16] N. Vaswani and W. Lu, "Modified-cs: Modifying compressive sensing for problems with partially known support," *IEEE Trans. Signal Process.*, vol. 58, no. 9, pp. 4595–4607, Sep. 2010.

[17] P. Pan, J. Feng, L. Chen, and Y. Yang, "Online compressed robust pca," in *International Joint Conference on Neural Networks (IJCNN)*, 2017, pp. 1041–1048.

[18] H. V. Luong, J. Seiler, A. Kaup, and S. Forchhammer, "Sparse signal reconstruction with multiple side information using adaptive weights for multiview sources," in *IEEE Int. Conf. on Image Process.*, Phoenix, Arizona, Sep. 2016.

[19] M. Brand, "Incremental singular value decomposition of uncertain data with missing values," in *European Conference on Computer Vision*, 2002.

[20] The matlab code of the CORPCA algorithm, the test sequences, and the corresponding outcomes. [Online]. Available: https://github.com/huynhlvd/corpca

[21] A. Beck and M. Teboulle, "A fast iterative shrinkage-thresholding algorithm for linear inverse problems," *SIAM Journal on Imaging Sciences*, vol. 2(1), pp. 183–202, 2009.

[22] J.-F. Cai, E. J. Candès, and Z. Shen, "A singular value thresholding algorithm for matrix completion," *SIAM J. on Optimization*, vol. 20, no. 4, pp. 1956–1982, Mar. 2010.

[23] J. F. Mota, N. Deligiannis, and M. R. Rodrigues, "Compressed sensing with side information: Geometrical interpretation and performance bounds," in *IEEE Global Conf. on Signal and Information Processing*, Austin, Texas, USA, Dec. 2014.

[24] ——, "Compressed sensing with prior information: Optimal strategies, geometry, and bounds," ArXiv e-prints, Aug. 2014.

[25] H. V. Luong, N. Deligiannis, J. Seiler, S. Forchhammer, and A. Kaup, "Incorporating prior information in compressive online robust principal component analysis," ArXiv, Jan. 2017.

[26] L. Li, W. Huang, I. Y.-H. Gu, and Q. Tian, "Statistical modeling of complex backgrounds for foreground object detection," *IEEE Trans. Image Process.*, vol. 13, no. 11, pp. 1459–1472, 2004.

[27] M. Dikmen, S. F. Tsai, and T. S. Huang, "Base selection in estimating sparse foreground in video," in *16th IEEE International Conference on Image Processing*, Nov 2009.