Compressive Online Robust Principal Component Analysis Via n-‘1 Minimization

Van Luong, Huynh; Deligiannis, Nikos; Seiler, Jurgen; Forchhammer, Soren; Kaup, Andre

Published in:
IEEE Transactions on Image Processing

Link to article, DOI:
10.1109/TIP.2018.2831915

Publication date:
2018

Document Version
Peer reviewed version

Compressive Online Robust Principal Component Analysis Via n-ℓ_1 Minimization

Huynh Van Luong, *Member, IEEE*, Nikos Deligiannis, *Member, IEEE*, Jürgen Seiler, *Member, IEEE*, Søren Forchhammer, *Member, IEEE*, and André Kaup, *Fellow, IEEE*

***Abstract*—This work considers online robust principal component analysis (RPCA) in time-varying 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. Different from conventional batch RPCA, which processes all the data directly, our approach considers a small set of measurements taken per data vector (frame). Moreover, our algorithm can incorporate multiple prior information from previous decomposed vectors via proposing an n-ℓ_1 minimization method. At each time instance, the algorithm recovers the sparse vector by solving the n-ℓ_1 minimization problem—which promotes not only the sparsity of the vector but also its correlation with multiple previously-recovered sparse vectors—and, subsequently, updates the low-rank component using incremental singular value decomposition. We also establish theoretical bounds on the number of measurements required to guarantee successful compressive separation under the assumptions of static or slowly-changing low-rank components. We evaluate the proposed algorithm using numerical experiments and online video foreground-background separation experiments. The experimental results show that the proposed method outperforms the existing methods.**

***Index Terms*—Robust PCA, compressed sensing, sparse signal, low-rank model, prior information.**

PRINCIPAL component analysis (PCA) [3] forms the crux of dimensionality reduction for high-dimensional data analysis. Robust PCA (RPCA) [4]–[7] has emerged as a counterpart to address the sensitivity of PCA to outliers and structured noise. A popular approach to RPCA decomposes a data matrix $\boldsymbol{M} \in \mathbb{R}^{n\times t}$ into the sum of a *sparse* component $\boldsymbol{X}$ and a *low-rank* component $\boldsymbol{L}$ by solving the principal component pursuit (PCP) [5] problem:

$$\min_{\boldsymbol{L},\boldsymbol{S}} \|\boldsymbol{L}\|_* + \lambda\|\text{vec}(\boldsymbol{X})\|_1 \text{ s.t. } \boldsymbol{M} = \boldsymbol{L} + \boldsymbol{X}, \qquad (1)$$

where $\|\boldsymbol{L}\|_* = \sum_i \sigma_i(\boldsymbol{L})$ is the nuclear norm—sum of singular values $\sigma_i(\boldsymbol{L})$—of the matrix $\boldsymbol{L}$, $\|\text{vec}(\boldsymbol{X})\|_1 = \sum_{i,j}|x_{i,j}|$ is the ℓ_1-norm of $\boldsymbol{X}$ organized in a vector, and λ is a tuning parameter. RPCA has found many applications in computer vision, web data analysis, anomaly detection, and data visualization (see [5], [8] and references therein). In video analysis, in particular, a sequence of vectorized frames (modeled by $\boldsymbol{M}$) is separated into the slowly-changing background $\boldsymbol{L}$ and the sparse foreground $\boldsymbol{X}$. Batch RPCA methods [4]–[7] typically assume that the low-dimensional subspace, where the data lives, is static; namely, the $\boldsymbol{L}$ component does not vary across the data samples. This assumption is typically invalid; even in static-camera video sequences, for example, the background might change due to illumination variations, moving water or leaves. In addition, these methods process all the data directly, leading to high computation and memory demands when high-dimensional streaming data such as video is analyzed.

In order to address these shortcomings, online RPCA methods [9]–[17] perform the decomposition recursively, i.e., per data vector (a.k.a., column-vector in $\boldsymbol{M}$). As such, online RPCA methods can process the incoming data vectors on-the-fly, without needing to store them in memory for computation at a later instance (as batch methods do). Several approaches to online RPCA have been proposed, including: (*i*) re-weighting each incoming data vector with a pre-defined influence function of its residual to the current estimated subspace [9]; (*ii*) performing matrix factorization of the low-rank component [10]; (*iii*) leveraging an exponentially-weighted least squares estimator [11]; (*iv*) assuming slow variation of the low-rank component and leveraging compressed sensing (CS) [12], [13] to recover the sparse component [14], [15]; and (*v*) applying alternating minimization to solve the PCP problem [16] and extending it to its incremental form [17]. Moreover, the online RPCA methods in [15], [18], [19] can operate on compressive measurements (a.k.a., subsampled or uncompleted data vectors), thereby reducing further the associated computation, communication and storage costs.

Problem definition. We consider a compressive online RPCA algorithm that recursively decomposes the data samples, observed through *low-dimensional measurements*, *by leveraging information from multiple previously decomposed data samples*. Formally, at time instance t, we wish to decompose $\boldsymbol{M}_t = \boldsymbol{L}_t + \boldsymbol{X}_t \in \mathbb{R}^{n\times t}$ into $\boldsymbol{X}_t = [\boldsymbol{x}_1\ \boldsymbol{x}_2\ ...\ \boldsymbol{x}_t]$ 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 $\boldsymbol{X}_t$ and $\boldsymbol{L}_t$, respectively. We assume that $\boldsymbol{L}_{t-1} = [\boldsymbol{v}_1\ \boldsymbol{v}_2\ ...\ \boldsymbol{v}_{t-1}]$ and $\boldsymbol{X}_{t-1} = [\boldsymbol{x}_1\ \boldsymbol{x}_2\ ...\ \boldsymbol{x}_{t-1}]$ 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 $\boldsymbol{\Phi} \in \mathbb{R}^{m\times n}$ $(m \ll n)$ is a random projection [12]. At time instance t, we formulate the decomposition problem

$$\begin{aligned} \min_{\boldsymbol{x}_t,\boldsymbol{v}_t} \quad & \|[\boldsymbol{L}_{t-1}\ \boldsymbol{v}_t]\|_* + \lambda_1\|\boldsymbol{x}_t\|_1 + \lambda_2 f(\boldsymbol{x}_t, \boldsymbol{X}_{t-1}) \qquad (2) \\ \text{s.t.} \quad & \boldsymbol{y}_t = \boldsymbol{\Phi}(\boldsymbol{x}_t + \boldsymbol{v}_t) \end{aligned}$$

Parts of this work have been presented at the IEEE ICIP 2016 [1] and the IEEE GlobalSIP 2017 [2].

H. V. Luong, J. Seiler, and A. Kaup are with the Chair of Multimedia Communications and Signal Processing, Friedrich-Alexander-Universität Erlangen-Nürnberg, 91058 Erlangen, Germany (e-mail: huynh.luong@fau.de, juergen.seiler@fau.de, andre.kaup@fau.de).

N. Deligiannis is with the Department of Electronics and Informatics, Vrije Universiteit Brussel, 1050 Brussels and with imec, Kapeldreef 75, 3001 Leuven, Belgium (e-mail: ndeligia@etrovub.be).

S. Forchhammer is with the Department of Photonics Engineering, Technical University of Denmark, 2800 Lyngby, Denmark (e-mail: sofo@fotonik.dtu.dk).

where $\boldsymbol{\Phi}$ is assumed known, $f(\cdot)$ is a function that expresses the similarity of $\boldsymbol{x}_t$ with the previously recovered sparse components $\boldsymbol{X}_{t-1}$, and λ_1, λ_2 are tuning parameters. Problem (2) emerges naturally in scenarios where the data samples have strong temporal correlation; for example, there is high correlation across the backgrounds and the foregrounds in multiple adjacent video frames [20], [21].

Relation to prior work. Unlike batch [4]–[7] or online [9]–[11], [17] RPCA approaches that consume full data samples, operating on compressive measurements (like our method does) improves the computational efficiency and reduces the data communication cost. Examples of compressive batch RPCA solutions include Compressive PCP [22] and SpaRCS [23]. The former solves a modified version of Problem (1) with constraint $\boldsymbol{D} = \boldsymbol{\Phi}(\boldsymbol{L} + \boldsymbol{X})$ using a fast interior-point algorithm [24], while the latter adheres to a greedy approach, combining CoSaMP [25] and ADMiRA [26] for sparse and low-rank recovery, respectively. Other studies have proposed online RPCA methods that consider sub-sampled data: GRASTA [18] alternates between an ADMM (alternating direction method of multipliers [27]) step to update the sparse component and an incremental gradient descent method on the Grassmannian manifold to update the low-rank component. GOSUS [28] improves over GRASTA by accounting for structured sparsity in the foreground, and ROSETA [29] by updating the sub-space using a proximal point iteration.

Recently, in the context of foreground-background separation, advanced factorization methods, which obtain robust background subtractions, have been proposed [30]–[32]. A low-rank matrix factorization method with a general noise distribution model was developed in [30]. Alternately, a Gaussian mixture model that is updated frame-by-frame and regularized by the previous foreground-background frames was developed in [31]. The method in [32] studied background subtraction from compressive measurements by proposing a tensor RPCA to exploit the spatial-temporal correlation in the background and the continuity of the foreground. The study in [19] presented an online compressive RPCA method that is supported by performance guarantees. However, these methods as well as the methods in [18], [28], [29] do not explore the correlation across adjacent sparse components, as we propose in Problem (2).

The problem of reconstructing a sequence of time-varying sparse signals using prior information has been explored in online RPCA [15] and in recursive CS [33]–[35] (see [36] for a review). The study in [37] extended the ReProCS [14], [15] algorithm to the compressive case and used modified-CS [38] to leverage prior support knowledge under the condition of slowly-varying support. However, no explicit condition on the number of measurements required for successful recovery was reported. Alternatively, the studies in [33]–[35], [39] assumed the low-rank vector non-varying ($\boldsymbol{v} = \boldsymbol{v}_1 = \cdots = \boldsymbol{v}_{t-1} = \boldsymbol{v}_t$) and recovered the sparse component using ℓ_1 [35], [39] or ℓ_1-ℓ_1 [33], [34] minimization. The studies in [33]–[35] also provide a method to compute the number of compressive measurements needed to recover $\boldsymbol{x}_t$ per time instance. However, the methods in [34], [35], [37] do not exploit multiple prior information from multiple previously recovered frames.

Contributions. We propose a compressive online RPCA (CORPCA) with multiple prior information algorithm that solves Problem (2). At each time instance, the algorithm recovers the sparse vector by minimizing an n-ℓ_1 norm minimization problem that promotes the sparsity of the vector and its correlation with multiple previously-recovered sparse vectors. To this end, we use a reconstruction algorithm with multiple side information using adaptive weights (RAMSIA) [1], our recent algorithm for compressed sensing with multiple prior information signals. Moreover, CORPCA updates the low-rank component via an incremental SVD [40] method, assuming that it varies slowly. Furthermore, we establish theoretical bounds on the number of measurements required by the proposed n-ℓ_1 minimization that serve as performance guarantees for CORPCA to obtain successful decomposition. We evaluate CORPCA using synthetic data as well as video data for foreground-background separation. The results show that the performance of CORPCA is following our theoretical analysis and it is higher than existing methods, that is, RPCA [5], GRASTA [18] and ReProCS [37].

Outline. The paper is continuous as follows. Section I presents the background of the work and Section II establishes our proposal for the function $f(\cdot)$ in (2) and elaborates on our algorithm to solve the n-ℓ_1 minimization problem. Section III gives the full description of the CORPCA method and derives its performance guarantees. Section IV describes our experimental results and Section V concludes the paper.

I. Background

We first review fundamental problems related to our work, namely, sparse signal recovery via ℓ_1 [12], [13] and ℓ_1-ℓ_1 minimization [41]–[43], and PCP [5]. Let $\boldsymbol{x} \in \mathbb{R}^n$ denote a finite-length, sparse or compressible signal for which we have access to low-dimensional, linear measurements $\boldsymbol{y} = \boldsymbol{\Phi}\boldsymbol{x} \in \mathbb{R}^m$, with $m \ll n$. The measurements are random projections obtained using a measurement matrix $\boldsymbol{\Phi} \in \mathbb{R}^{m \times n}$, whose elements are sampled from an i.i.d. Gaussian distribution. According to the compressed sensing (CS) theory [12], [13], $\boldsymbol{x}$ can be recovered with overwhelming probability by solving the following ℓ_1 minimization problem:

$$\min_{\boldsymbol{x}} ||\boldsymbol{x}||_1 \text{ s.t. } \boldsymbol{y} = \boldsymbol{\Phi}\boldsymbol{x}, \tag{3}$$

provided that [44]

$$m_{\ell_1} \geq 2s_0 \log(n/s_0) + (7/5)s_0 + 1, \tag{4}$$

where $s_0 := \|\boldsymbol{x}\|_0 = |\{i : x_i \neq 0\}|$ denotes the number of nonzero elements in $\boldsymbol{x}$, with $|\cdot|$ being the cardinality of a set and $\|\cdot\|_0$ the ℓ_0 pseudo-norm. Problem (3) can be solved by a relaxed problem as

$$\min_{\boldsymbol{x}} \{H(\boldsymbol{x}) = f(\boldsymbol{x}) + g(\boldsymbol{x})\}, \tag{5}$$

where $f := \mathbb{R}^n \to \mathbb{R}$ is a smooth convex function with Lipschitz constant $L_{\nabla f}$ [45] of the gradient ∇f and $g := \mathbb{R}^n \to \mathbb{R}$ is a continuous convex function possibly non-smooth. Problem (3) arises from (5) when $g(\boldsymbol{x}) = \lambda||\boldsymbol{x}||_1$, with $\lambda > 0$ being a regularization parameter, and $f(\boldsymbol{x}) = \frac{1}{2}||\boldsymbol{\Phi}\boldsymbol{x} - \boldsymbol{y}||_2^2$. By using a proximal gradient method [45], $\boldsymbol{x}^{(k)}$ at iteration k can be iteratively computed as

$$\boldsymbol{x}^{(k)} = \Gamma_{\frac{1}{L}g}\Big(\boldsymbol{x}^{(k-1)} - \frac{1}{L}\nabla f(\boldsymbol{x}^{(k-1)})\Big), \tag{6}$$

where $L \geq L_{\nabla f}$, and $\Gamma_{\frac{1}{L}g}(\boldsymbol{x})$ is a proximal operator defined as

$$\Gamma_{\frac{1}{L}g}(\boldsymbol{x}) = \arg\min_{\boldsymbol{v}\in\mathbb{R}^n} \left\{ \frac{1}{L} g(\boldsymbol{v}) + \frac{1}{2}||\boldsymbol{v}-\boldsymbol{x}||_2^2 \right\}. \quad (7)$$

The performance of CS can be improved by leveraging a signal $\boldsymbol{z}$ correlated to the target signal $\boldsymbol{x}$—called prior (or side) information—which is available at the reconstruction side [38], [41], [42]. The study in [41], [42] showed that by solving an ℓ_1-ℓ_1 minimization problem the performance of CS can be improved dramatically under the condition that the prior information has good enough quality [41], [42]. The ℓ_1-ℓ_1 minimization problem attempts to reconstruct $\boldsymbol{x}$ given a side information signal $\boldsymbol{z} \in \mathbb{R}^n$ by solving Problem (5) with $g(\boldsymbol{x}) = \lambda(\|\boldsymbol{x}\|_1 + \|\boldsymbol{x}-\boldsymbol{z}\|_1)$, that is,

$$\min_{\boldsymbol{x}} \left\{ \frac{1}{2}\|\boldsymbol{\Phi}\boldsymbol{x}-\boldsymbol{y}\|_2^2 + \lambda(\|\boldsymbol{x}\|_1 + \|\boldsymbol{x}-\boldsymbol{z}\|_1) \right\}. \quad (8)$$

The study in [41], [42] derived the following bound on the number of measurements required for Problem (8) to successfully reconstruct $\boldsymbol{x}$:

$$m_{\ell_1\text{-}\ell_1} \geq 2\bar{h}\log\left(\frac{n}{s_0+\xi/2}\right) + \frac{7}{5}\left(s_0+\frac{\xi}{2}\right)+1, \quad (9)$$

where ξ, $\bar{h}$ are quantities—expressing the correlation between the target signal and the prior information—defined as

$$\xi := |\{i : z_i \neq x_i = 0\}| - |\{i : z_i = x_i \neq 0\}|, \quad (10a)$$

$$\bar{h} := |\{i : x_i > 0, x_i > z_i\} \cup \{i : x_i < 0, x_i < z_i\}|, \quad (10b)$$

wherein x_i, z_i are the i-th elements in $\boldsymbol{x}$, $\boldsymbol{z}$.

The PCP [5] problem defined in (1) subsumes the CS problem. To show this, we follow the formulation in (5) and write Problem (1) as [4]

$$\min_{\boldsymbol{L},\boldsymbol{X}} \{H(\boldsymbol{L},\boldsymbol{X}) = \mathcal{F}(\boldsymbol{L},\boldsymbol{X}) + \mathcal{G}(\boldsymbol{L},\boldsymbol{X})\}, \quad (11)$$

where $\mathcal{F}(\boldsymbol{L},\boldsymbol{X}) = \frac{1}{2}\|\boldsymbol{M}-\boldsymbol{L}-\boldsymbol{X}\|_F^2$ and $\mathcal{G}(\boldsymbol{L},\boldsymbol{X}) = \mu\|\boldsymbol{L}\|_* + \mu\lambda\|\boldsymbol{X}\|_1$, with $\|\cdot\|_F$ denoting the Frobenious norm. The solution of (11) using proximal gradient methods [45] gives that $\boldsymbol{L}^{(k+1)}$ and $\boldsymbol{X}^{(k+1)}$ at iteration $k+1$ can be iteratively computed via the singular value thresholding operator [46] for $\boldsymbol{L}$ and the soft thresholding operator [45] for $\boldsymbol{X}$

$$\boldsymbol{L}^{(k+1)} = \arg\min_{\boldsymbol{L}} \Big\{ \mu\|\boldsymbol{L}\|_* + \Big\| \boldsymbol{L} - \Big(\boldsymbol{L}^{(k)} - \frac{1}{2}\nabla_{\boldsymbol{L}}\mathcal{F}(\boldsymbol{L}^{(k)},\boldsymbol{X}^{(k)}) \Big) \Big\|_F^2 \Big\} \quad (12a)$$

$$\boldsymbol{X}^{(k+1)} = \arg\min_{\boldsymbol{X}} \Big\{ \mu\lambda\|\boldsymbol{X}\|_1 + \Big\| \boldsymbol{X} - \Big(\boldsymbol{X}^{(k)} - \frac{1}{2}\nabla_{\boldsymbol{X}}\mathcal{F}(\boldsymbol{L}^{(k)},\boldsymbol{X}^{(k)}) \Big) \Big\|_F^2 \Big\}. \quad (12b)$$

II. Sparse Signal Recovery with Multiple Prior Information

A. Problem Statement

We consider the problem of reconstructing efficiently a sparse signal given multiple prior signals; specifically, we study how to leverage—and adapt to—the varying correlations among a target sparse signal and multiple known prior signals. Let us consider the application of video reconstruction: Figs. 1(a)-1(c) depict three consecutive frames from the `Bootstrap` [47] video sequence and Figs. 1(d), 1(f), 1(h) depict the corresponding foregrounds after performing manual separation. Each foreground signal is vectorized (in the form of a column-vector in $\boldsymbol{X} = [\ldots, \boldsymbol{x}_{t-2}, \boldsymbol{x}_{t-1}, \boldsymbol{x}_t, \ldots]$) and is depicted in Figs. 1(e), 1(g), and 1(i), respectively. It is clear that (*i*) each foreground signal $\boldsymbol{x}_t$ is sparse; and (*ii*) there is high (temporal) correlation across the consecutive foreground signals. Suppose that we have access to a low-dimensional measurement vector $\boldsymbol{y}_t = \boldsymbol{\Phi}\boldsymbol{x}_t$ from the foreground at time instance t (corresponding, for example, to frame no. 1286). Since $\boldsymbol{x}_t$ is sparse we can recover it by solving Problem (3) provided that the dimension of $\boldsymbol{y}_t$ adheres to (4) [this was the main concept behind the compressive foreground extraction work in [39], where it was assumed that the background is constant]. Alternatively, we can attempt to recover $\boldsymbol{x}_t$ by solving the ℓ_1-ℓ_1 minimization problem in (8) [41], [42], where a previously recovered signal (for example, $\boldsymbol{x}_{t-1}$ or $\boldsymbol{x}_{t-2}$) can play the role of the prior information signal $\boldsymbol{z}$ [this was in turn the idea behind the work in [33], [34], where as in [39] it was assumed that the background is non-varying]. It might however be that the ℓ_1-ℓ_1 minimization method performs worse than ℓ_1 minimization due to poor correlation between the prior signal $\boldsymbol{z}$ and the target signal $\boldsymbol{x}_t$ [41], [42]. To address this issue, we propose RAMSIA a new sparse signal reconstruction algorithm that uses adaptively-learned weights to balance the contributions from multiple prior information signals. RAMSIA formulates a n-ℓ_1 minimization problem the cost function of which is obtained by replacing the following function in (5):

$$g(\boldsymbol{x}) = \lambda\Big(\beta_0\|\mathbf{W}_0\boldsymbol{x}\|_1 + \sum_{j=1}^{J}\beta_j\|\mathbf{W}_j(\boldsymbol{x}-\boldsymbol{z}_j)\|_1\Big), \quad (13)$$

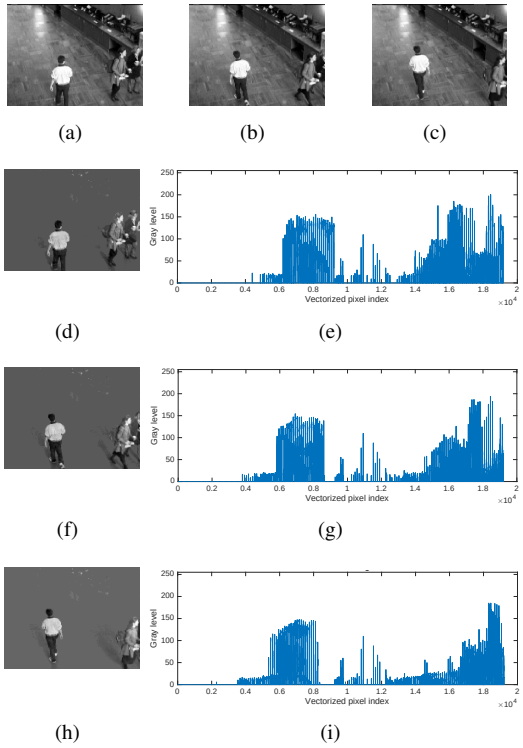


Fig. 1. Consecutive frames in the `Bootstrap` video sequence [frame no. 1284, 1285, 1286 in (a), (b), (c), respectively], as well as the depictions [(d), (f), (h)] and vectorized pixel values [(e), (g), (i)] of the corresponding foregrounds.

where $\boldsymbol{x}$ (a.k.a., $\boldsymbol{x}_t$) is the signal to be recovered, $\boldsymbol{z}_1, \ldots, \boldsymbol{z}_J$ are J prior information signals (a.k.a., $\boldsymbol{z}_j$ corresponds to $\boldsymbol{x}_{t-j}$, $j \in \{1, \ldots, J\}$), $\beta_j > 0$ are weights across the prior information vectors, and the element-wise weights, $\mathbf{W}_j = \text{diag}(w_{j1}, w_{j2}, ..., w_{jn})$, are assigned to each element $x_i - z_{ji}$ in the vector $\boldsymbol{x} - \boldsymbol{z}_j$. By writing the first summation term in (13) as $\beta_0 \|\mathbf{W}_0(\boldsymbol{x} - \boldsymbol{z}_0)\|_1$ with $\boldsymbol{z}_0 = \mathbf{0}$, we can then compact the function as

$$g(\boldsymbol{x}) = \lambda \sum_{j=0}^{J} \beta_j \|\mathbf{W}_j(\boldsymbol{x} - \boldsymbol{z}_j)\|_1. \tag{14}$$

We end up formulating the objective function of the n-ℓ_1 problem as

$$\min_{\boldsymbol{x}} \Big\{ H(\boldsymbol{x}) = \frac{1}{2} ||\boldsymbol{\Phi}\boldsymbol{x} - \boldsymbol{y}||_2^2 + \lambda \sum_{j=0}^{J} \beta_j ||\mathbf{W}_j(\boldsymbol{x} - \boldsymbol{z}_j)||_1 \Big\}. \tag{15}$$

The use of multiple prior information signals is the main novelty of our algorithm compared to prior studies. Furthermore, the use of weights across the side information vectors as well as the elements of each vector differentiates RAMSIA from our previous work [48]. We note that (15) boils down to Problem (8) when $\beta_j = 1$, $\mathbf{W}_0 = \mathbf{W}_1 = \mathbf{I}$, and $\mathbf{W}_j = \mathbf{0}$ $(j \geq 2)$, where $\mathbf{I}$ is the $n \times n$ identity matrix.

B. The proposed RAMSIA Algorithm

We solve the n-ℓ_1 minimization problem in (15) by modifying the fast iterative soft thresholding (FISTA) method [45], where, at every iteration k of the algorithm, we update the weights β_j and $\mathbf{W}_j$, and also compute $\boldsymbol{x}$. In this way, we adaptively weight multiple prior information signals according to their qualities during the iterative process. We may have different strategies to update the weights depending on our constraints. In this work[1], we set the constraints as $\sum_{i=1}^{n} w_{ji} = n$ and $\sum_{j=0}^{J} \beta_j = 1$.

At each iteration, our algorithm proceeds as follows: Firstly, we keep $\mathbf{W}_j$ and β_j fixed, and we compute $\boldsymbol{x}$ by minimizing Problem (15). Adhering to the proximal gradient method [45], $\boldsymbol{x}^{(k)}$ at iteration k is obtained from (6), where $g(\boldsymbol{x}) = \lambda \sum_{j=0}^{J} \beta_j \|\mathbf{W}_j(\boldsymbol{x} - \boldsymbol{z}_j)\|_1$. The proximal operator $\Gamma_{\frac{1}{L}g}(\boldsymbol{x})$ (7) for our problem is given by [we restate here the result derived in Proposition A.1 in Appendix A]

$$\Gamma_{\frac{1}{L}g}(x_i) = \begin{cases} x_i - \frac{\lambda}{L} \sum_{j=0}^{J} \beta_j w_{ji} (-1)^{\mathfrak{b}(l<j)} & \text{if (17a)}, \\ z_{li} & \text{if (17b)}, \end{cases} \tag{16}$$

with

$$z_{li} + \frac{\lambda}{L} \sum_{j=0}^{J} \beta_j w_{ji} (-1)^{\mathfrak{b}(l<j)} < x_i < z_{(l+1)i} + \frac{\lambda}{L} \sum_{j=0}^{J} \beta_j w_{ji} (-1)^{\mathfrak{b}(l<j)}, \tag{17a}$$

$$z_{li} + \frac{\lambda}{L} \sum_{j=0}^{J} \beta_j w_{ji} (-1)^{\mathfrak{b}(l-1<j)} \leq x_i \leq z_{li} + \frac{\lambda}{L} \sum_{j=0}^{J} \beta_j w_{ji} (-1)^{\mathfrak{b}(l<j)}, \tag{17b}$$

where, without loss of generality, we have assumed that $-\infty = z_{(-1)i} \leq z_{0i} \leq z_{1i} \leq \ldots \leq z_{Ji} \leq z_{(J+1)i} = \infty$, and we have defined a boolean function

$$\mathfrak{b}(l < j) = \begin{cases} 1, & \text{if} \quad l < j \\ 0, & \text{otherwise.} \end{cases} \tag{18}$$

with $l \in \{-1, \ldots, J\}$. It is worth noting that (17a) and (17b) are disjoint.

Secondly, given $\boldsymbol{x}$ and β_j, we update $\mathbf{W}_j$ per prior information $\boldsymbol{z}_j$. In order to improve the reconstruction of $\boldsymbol{x}$, we aim at assigning each element $x_i - z_{ji}$ in the vector $\boldsymbol{x} - \boldsymbol{z}_j$ to a weight w_{ji} that is higher when $|x_i - z_{ji}|$ is low. In this way, we can promote the sparsity of the solution of the n-ℓ_1 minimization problem[2]. Therefore, we set $w_{ji} = \eta/(|x_i - z_{ji}| + \epsilon)$ for all $i = \{1, \ldots, n\}$, where a small parameter $\epsilon > 0$ is added to ensure that zero-valued elements $|x_i - z_{ji}|$ do not cause a breakdown of the method. Setting the constraint $\sum_{i=1}^{n} w_{ji} = n$, i.e., $\sum_{i=1}^{n} (\eta/(|x_i - z_{ji}| + \epsilon)) = n$, we derive

$$\eta = \frac{n}{\sum_{l=1}^{n} (|x_i - z_{jl}| + \epsilon)^{-1}}. \tag{19}$$

Then, we get

$$w_{ji} = \frac{n(|x_i - z_{ji}| + \epsilon)^{-1}}{\sum_{l=1}^{n} (|x_i - z_{jl}| + \epsilon)^{-1}}. \tag{20}$$

Thirdly, keeping $\boldsymbol{x}$ and $\mathbf{W}_j$ fixed, we compute β_j across the prior information vectors—which should have a higher value when they correspond to priors $\boldsymbol{z}_j$ closer to $\boldsymbol{x}$—similar to w_{ji}, we obtain β_j as

$$\beta_j = \eta/(\|\mathbf{W}_j(\boldsymbol{x} - \boldsymbol{z}_j)\|_1 + \epsilon). \tag{21}$$

Combining (21) with the constraint $\sum_{j=0}^{J} \beta_j = 1$, similar to the derivation of w_{ji} in (20), we get

$$\beta_j = \frac{\Big(\|\mathbf{W}_j(\boldsymbol{x} - \boldsymbol{z}_j)\|_1 + \epsilon\Big)^{-1}}{\sum_{l=0}^{J} \Big(\|\mathbf{W}_l(\boldsymbol{x} - \boldsymbol{z}_l)\|_1 + \epsilon\Big)^{-1}}. \tag{22}$$

The steps of RAMSIA[3] are summarized in Algorithm 1. Its main difference w.r.t. FISTA [45] is the iterative updating of the weights using (20) and (22). The stopping criterion for Algorithm 1 can be either a maximum number of iterations $k_{\max}$ reached, a small relative variation of the objective function $H(\boldsymbol{x})$ in (15), or a small change of the number of nonzero components of the estimate $\boldsymbol{x}^{(k)}$ at iteration k. In this work, we use the small relative variation of $H(\boldsymbol{x})$.

III. The Proposed CORPCA Method

We now present our compressive online RPCA (CORPCA) algorithm and we derive a bound on the number of required measurements that guarantee successful signal decomposition.

[1] In [48], we used a different weighting strategy; specifically, $\sum_{j=0}^{J} w_{ji} = 1$ and $\beta_j = 1$, $\forall j = \{0, 1, \ldots . J\}$.

[2] The ℓ_1 minimization is a relaxation of the ℓ_0 minimization problem for recovering a sparse signal. The ℓ_1 norm depends on the magnitude of the nonzero signal coefficients, while the ℓ_0 norm only counts the number nonzero coefficients in the signal. Hence, as proposed in [49], the weights in the re-weighted version of ℓ_1 minimization are designed to reduce the impact of the magnitude of the nonzero elements, thereby leading to a solution that approximates better the one obtained with ℓ_0 minimization. For further details on re-weighted ℓ_1 minimization, we refer to [49].

[3] The code is available online at https://github.com/huynhlvd/ramsia.

Algorithm 1: The proposed RAMSIA algorithm.

Input: $\boldsymbol{y},\ \boldsymbol{\Phi},\ \boldsymbol{z}_1,\ \boldsymbol{z}_2, ...,\ \boldsymbol{z}_J$;
Output: $\widehat{\boldsymbol{x}}$;
// Initialization.
1 $\boldsymbol{u}^{(1)}=\boldsymbol{x}^{(0)}=\mathbf{0}$; $\mathbf{W}_0=\mathbf{I}$; $\beta_0=1$; $\mathbf{W}_j=\mathbf{0}$; $\beta_j=0\ (1\leq j\leq J)$; $L=L_{\nabla f}$; $\lambda, \epsilon>0$; $\xi_1=1$; $k=0$;
2 **while** *stopping criterion is false* **do**
3 $k=k+1$;
// Solving (15) given the weights.
4 $\nabla f(\boldsymbol{u}^{(k)})=\boldsymbol{\Phi}^{\mathrm{T}}(\boldsymbol{\Phi}\boldsymbol{u}^{(k)}-\boldsymbol{y})$;
5 $\boldsymbol{x}^{(k)}=\Gamma_{\frac{1}{L}g}\Big(\boldsymbol{u}^{(k)}-\frac{1}{L}\nabla f(\boldsymbol{u}^{(k)})\Big)$; where $\Gamma_{\frac{1}{L}g}(\cdot)$ is given by (16);
// Computing the updated weights.
6 $w_{ji}=\dfrac{n(|x_i^{(k)}-z_{ji}|+\epsilon)^{-1}}{\sum_{l=1}^{n}(|x_i^{(k)}-z_{jl}|+\epsilon)^{-1}}$;
7 $\beta_j=\dfrac{\Big(||\mathbf{W}_j(\boldsymbol{x}^{(k)}-\boldsymbol{z}_j)||_1+\epsilon\Big)^{-1}}{\sum_{l=0}^{J}\Big(||\mathbf{W}_l(\boldsymbol{x}^{(k)}-\boldsymbol{z}_l)||_1+\epsilon\Big)^{-1}}$;
// Updating new values.
8 $\xi_{k+1}=(1+\sqrt{1+4\xi_k^2})/2$;
9 $\boldsymbol{u}^{(k+1)}=\boldsymbol{x}^{(k)}+\frac{\xi_k-1}{\xi_{k+1}}(\boldsymbol{x}^{(k)}-\boldsymbol{x}^{(k-1)})$;
10 **end**
11 **return** $\widehat{\boldsymbol{x}}=\boldsymbol{x}^{(k)}$;

A. The CORPCA Objective Function

At time instance t, the algorithm receives as input a random projection $\boldsymbol{y}_t=\boldsymbol{\Phi}_{\mathbf{t}}(\boldsymbol{x}_t+\boldsymbol{v}_t)\in\mathbb{R}^{m_t}$ of a data vector (a.k.a., video frame) and estimates the sparse and low-rank components ($\widehat{\boldsymbol{x}}_t$, $\widehat{\boldsymbol{v}}_t$, respectively) with the aid of prior information gleaned from multiple previously decomposed vectors. The algorithm solves the following optimization problem:

$$\min_{\boldsymbol{x}_t,\boldsymbol{v}_t}\Big\{H(\boldsymbol{x}_t,\boldsymbol{v}_t)=\frac{1}{2}\|\boldsymbol{\Phi}_{\mathbf{t}}(\boldsymbol{x}_t+\boldsymbol{v}_t)-\boldsymbol{y}_t\|_2^2 + \lambda\mu\sum_{j=0}^{J}\beta_j\|\mathbf{W}_j(\boldsymbol{x}_t-\boldsymbol{z}_j)\|_1+\mu\Big\|[\boldsymbol{B}_{t-1}\ \boldsymbol{v}_t]\Big\|_*\Big\}, \quad (23)$$

where λ, $\mu>0$ are tuning parameters, and $\boldsymbol{Z}_{t-1}:=\{\boldsymbol{z}_j\}_{j=1}^{J}$, $\boldsymbol{B}_{t-1}\in\mathbb{R}^{n\times d}$ denotes a matrix, of which serve as prior information for $\boldsymbol{x}_t$ and $\boldsymbol{v}_t$, respectively. The components in $\boldsymbol{Z}_{t-1}$ and $\boldsymbol{B}_{t-1}$ can be a direct (sub-)set of the previously reconstructed data vectors $\{\widehat{\boldsymbol{x}}_1,...,\widehat{\boldsymbol{x}}_{t-1}\}$ and $\{\widehat{\boldsymbol{v}}_1,...,\widehat{\boldsymbol{v}}_{t-1}\}$, or formed after applying a processing step. In the case of video data, the processing step can compensate for the motion across the frames, for example, by means of optical flow [50]. Finally, the reconstructed vectors $\widehat{\boldsymbol{x}}_t$ and $\widehat{\boldsymbol{v}}_t$ are used to update the prior information $\boldsymbol{Z}_t$ and $\boldsymbol{B}_t$, which is used at the decomposition at time instance $t+1$.

We note that Problem (23) is obtained by replacing the terms $\lambda_1\|\boldsymbol{x}_t\|_1+\lambda_2 f(\boldsymbol{x}_t,\boldsymbol{X}_{i-1})$ in Problem (2) with the function $g(\boldsymbol{x})$ in the objective function of RAMSIA [c.f. (14)]. It is also worth observing that when $\boldsymbol{v}_t$ is non-varying, Problem (23) becomes Problem (15). 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 $\boldsymbol{\Phi}_{\mathbf{t}}$ into account, Problem (23) becomes Problem (1).

B. The CORPCA Algorithm

The proposed CORPCA[4] algorithm operates in two steps: It first solves Problem (23) given $\boldsymbol{Z}_{t-1}$ and $\boldsymbol{B}_{t-1}$ and then updates the matrices $\boldsymbol{Z}_t$ and $\boldsymbol{B}_t$, which are to be used in the following time instance.

Solution of Problem (23). Let us denote $f(\boldsymbol{v}_t,\boldsymbol{x}_t)=(1/2)\|\boldsymbol{\Phi}_{\mathbf{t}}(\boldsymbol{x}_t+\boldsymbol{v}_t)-\boldsymbol{y}_t\|_2^2$, $g(\boldsymbol{x}_t)=\lambda\sum_{j=0}^{J}\beta_j\|\mathbf{W}_j(\boldsymbol{x}_t-\boldsymbol{z}_j)\|_1$, and $h(\boldsymbol{v}_t)=\|[\boldsymbol{B}_{t-1}\ \boldsymbol{v}_t]\|_*$. As shown in Algorithm 2, CORPCA solves (23) by using proximal gradient methods [4], [45]. Specifically, as shown in Lines 3-9 in Algorithm 2, the algorithm computes $\boldsymbol{x}_t^{(k+1)}$ and $\boldsymbol{v}_t^{(k+1)}$ at iteration $k+1$ via the soft thresholding operator [45] and the singular value thresholding operator [46], respectively:

$$\boldsymbol{x}_t^{(k+1)}=\arg\min_{\boldsymbol{x}_t}\Big\{\mu g(\boldsymbol{x}_t)+\Big\|\boldsymbol{x}_t-\Big(\boldsymbol{x}_t^{(k)}-\frac{1}{2}\nabla_{\boldsymbol{x}_t}f(\boldsymbol{v}_t^{(k)},\boldsymbol{x}_t^{(k)})\Big)\Big\|_2^2\Big\}. \quad (24)$$

$$\boldsymbol{v}_t^{(k+1)}=\arg\min_{\boldsymbol{v}_t}\Big\{\mu h(\boldsymbol{v}_t)+\Big\|\boldsymbol{v}_t-\Big(\boldsymbol{v}_t^{(k)}-\frac{1}{2}\nabla_{\boldsymbol{v}_t}f(\boldsymbol{v}_t^{(k)},\boldsymbol{x}_t^{(k)})\Big)\Big\|_2^2\Big\}, \quad (25)$$

The proximal operator $\boldsymbol{\Gamma}_{\tau g_1}(\cdot)$ in Line 7 of Algorithm 2 is defined as

$$\boldsymbol{\Gamma}_{\tau g_1}(\boldsymbol{X})=\arg\min_{\boldsymbol{V}}\Big\{\tau g_1(\boldsymbol{V})+\frac{1}{2}||\boldsymbol{V}-\boldsymbol{X}||_F^2\Big\}, \quad (26)$$

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 [4], the convergence of the convex problem (23) of Algorithm 2 in Line 2 is determined by evaluating the criterion $\|\partial H(\boldsymbol{x}_t,\boldsymbol{v}_t)|_{\boldsymbol{x}_t^{(k+1)},\boldsymbol{v}_t^{(k+1)}}\|_2^2<2\times10^{-7}\|(\boldsymbol{x}_t^{(k+1)},\boldsymbol{v}_t^{(k+1)})\|_2^2$.

Prior Update. Updating $\boldsymbol{Z}_t$ and $\boldsymbol{B}_t$ is carried out after each time instance (see Lines 15-16 in Algorithm 2). Due to the correlation between subsequent signals (a.k.a., video frames in our example), 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 value decomposition (SVD) [40] method [referring to the function $\mathrm{incSVD}(\cdot)$ in Line 6 of Algorithm 2]. 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 time instance. However, in order to maintain a reasonable number of d, we take $\boldsymbol{B}_t=\boldsymbol{U}_t(:,1:d)\boldsymbol{\Gamma}_{\frac{\mu_k}{2}g_1}(\boldsymbol{\Sigma}_t)(1:d,1:d)\boldsymbol{V}_t^{\mathrm{T}}(:,1:d)$. The computational cost of $\mathrm{incSVD}(\cdot)$ is lower than that of the conventional SVD [17], [40] 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 $\mathrm{incSVD}(\cdot)$ is presented in the following: The goal is to compute $\mathrm{incSVD}[\boldsymbol{B}_{t-1}\ \boldsymbol{v}_t]$, 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 the components $(\boldsymbol{U}_t,\boldsymbol{\Sigma}_t,\boldsymbol{V}_t)$ using the components $(\boldsymbol{U}_{t-1},\boldsymbol{\Sigma}_{t-1},\boldsymbol{V}_{t-1})$ and the vector $\boldsymbol{v}_t$. We write the matrix $[\boldsymbol{B}_{t-1}\ \boldsymbol{v}_t]$ as

[4]The code for CORPCA is available online [51].

Algorithm 2: The proposed CORPCA algorithm.

Input: $\boldsymbol{y}_t,\ \boldsymbol{\Phi_t},\ \boldsymbol{Z}_{t-1},\ \boldsymbol{B}_{t-1}$;
Output: $\widehat{\boldsymbol{x}}_t,\ \widehat{\boldsymbol{v}}_t,\ \boldsymbol{Z}_t,\ \boldsymbol{B}_t$;
// Initialize variables and parameters.
$\boldsymbol{x}_t^{(-1)}=\boldsymbol{x}_t^{(0)}=\mathbf{0}$; $\boldsymbol{v}_t^{(-1)}=\boldsymbol{v}_t^{(0)}=\mathbf{0}$; $\mathbf{W}_0=\mathbf{I}$; $\beta_0=1$; $\mathbf{W}_j=\mathbf{0}$; $\beta_j=0$ $(1\le j\le J)$; $\xi_{-1}=\xi_0=1$; $\mu_0=0$; $\bar{\mu}>0$; $\lambda>0$; $0<\epsilon<1$; $k=0$; $g_1(\cdot)=\|\cdot\|_1$;
while *not converged* **do**
// Solve Problem (23).
$\widetilde{\boldsymbol{v}}_t^{(k)}=\boldsymbol{v}_t^{(k)}+\frac{\xi_{k-1}-1}{\xi_k}(\boldsymbol{v}_t^{(k)}-\boldsymbol{v}_t^{(k-1)})$;
$\widetilde{\boldsymbol{x}}_t^{(k)}=\boldsymbol{x}_t^{(k)}+\frac{\xi_{k-1}-1}{\xi_k}(\boldsymbol{x}_t^{(k)}-\boldsymbol{x}_t^{(k-1)})$;
$\nabla_{\boldsymbol{v}_t} f(\widetilde{\boldsymbol{v}}_t^{(k)},\widetilde{\boldsymbol{x}}_t^{(k)})=\nabla_{\boldsymbol{x}_t} f(\widetilde{\boldsymbol{v}}_t^{(k)},\widetilde{\boldsymbol{x}}_t^{(k)})=\boldsymbol{\Phi_t}^{\mathrm{T}}\Big(\boldsymbol{\Phi_t}(\widetilde{\boldsymbol{v}}_t^{(k)}+\widetilde{\boldsymbol{x}}_t^{(k)})-\boldsymbol{y}_t\Big)$;
$(\boldsymbol{U}_t,\boldsymbol{\Sigma}_t,\boldsymbol{V}_t)=\mathrm{incSVD}\Big(\Big[\boldsymbol{B}_{t-1}\ \Big(\widetilde{\boldsymbol{v}}_t^{(k)}-\frac{1}{2}\nabla_{\boldsymbol{v}_t} f(\widetilde{\boldsymbol{v}}_t^{(k)},\widetilde{\boldsymbol{x}}_t^{(k)})\Big)\Big]\Big)$;
$\boldsymbol{\Theta}_t=\boldsymbol{U}_t\boldsymbol{\Gamma}_{\frac{\mu_k}{2}g_1}(\boldsymbol{\Sigma}_t)\boldsymbol{V}_t^T$;
$\boldsymbol{v}_t^{(k+1)}=\boldsymbol{\Theta}_t(:,\mathrm{end})$;
$\boldsymbol{x}_t^{(k+1)}=\Gamma_{\frac{\mu_k}{2}g}\Big(\widetilde{\boldsymbol{x}}_t^{(k)}-\frac{1}{2}\nabla_{\boldsymbol{x}_t} f(\widetilde{\boldsymbol{v}}_t^{(k)},\widetilde{\boldsymbol{x}}_t^{(k)})\Big)$; where $\Gamma_{\frac{\mu_k}{2}g}(\cdot)$ is given by (16);
// Compute the updated weights.
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 $k=k+1$;
end
// Update prior information.
15 $\boldsymbol{Z}_t:=\{\boldsymbol{z}_j=\boldsymbol{x}_{t-J+j}^{(k+1)}\}_{j=1}^{J}$;
16 $\boldsymbol{B}_t=\boldsymbol{U}_t(:,1:d)\boldsymbol{\Gamma}_{\frac{\mu_k}{2}g_1}(\boldsymbol{\Sigma}_t)(1:d,1:d)\boldsymbol{V}_t^{\mathrm{T}}(:,1:d)$;
17 **return** $\widehat{\boldsymbol{x}}_t=\boldsymbol{x}_t^{(k+1)},\ \widehat{\boldsymbol{v}}_t=\boldsymbol{v}_t^{(k+1)},\ \boldsymbol{Z}_t,\ \boldsymbol{B}_t$;

$$[\boldsymbol{B}_{t-1}\ \boldsymbol{v}_t]=\Big[\boldsymbol{U}_{t-1}\ \frac{\boldsymbol{\delta}_t}{\|\boldsymbol{\delta}_t\|_2}\Big]\cdot\begin{bmatrix}\boldsymbol{\Sigma}_{t-1} & \boldsymbol{e}_t\\ \mathbf{0}^{\mathrm{T}} & \|\boldsymbol{\delta}_t\|_2\end{bmatrix}\cdot\begin{bmatrix}\boldsymbol{V}_{t-1}^{\mathrm{T}} & \mathbf{0}\\ \mathbf{0}^{\mathrm{T}} & 1\end{bmatrix}, \tag{27}$$

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 SVD of the matrix in between the right side of (27), we have $\begin{bmatrix}\boldsymbol{\Sigma}_{t-1} & \boldsymbol{e}_t\\ \mathbf{0}^{\mathrm{T}} & \|\boldsymbol{\delta}_t\|_2\end{bmatrix}=\widetilde{\boldsymbol{U}}\widetilde{\boldsymbol{\Sigma}}\widetilde{\boldsymbol{V}}^{\mathrm{T}}$. Finally, we obtain $\boldsymbol{U}_t=\Big[\boldsymbol{U}_{t-1}\ \frac{\boldsymbol{\delta}_t}{\|\boldsymbol{\delta}_t\|_2}\Big]\cdot\widetilde{\boldsymbol{U}}$, $\boldsymbol{\Sigma}_t=\widetilde{\boldsymbol{\Sigma}}$, and $\boldsymbol{V}_t=\begin{bmatrix}\boldsymbol{V}_{t-1}^{\mathrm{T}} & \mathbf{0}\\ \mathbf{0}^{\mathrm{T}} & 1\end{bmatrix}\cdot\widetilde{\boldsymbol{V}}$.

C. Performance Guarantees

We establish a bound for the minimum number of measurements required to solve the n-ℓ_1 minimization in Problem (23). The bound depends on the support of the sparse vector $\boldsymbol{x}_t$ to be recovered and the correlations between $\boldsymbol{x}_t$ and the multiple priors $\boldsymbol{z}_j$. The correlations are expressed via the supports of the differences $\boldsymbol{x}_t-\boldsymbol{z}_j$. The bound serves as performance guarantees on the number of measurements for COPRCA, which solves iteratively the n-ℓ_1 minimization problem (23). This bound is useful for adaptively determining the compressive rate for recursive separation; namely, at a given time instance, the bound can be used to predict the number of measurements required at the next time instance. In our theoretical analysis, we assume that $\boldsymbol{v}_t$ is slowly-changing incurring a measurement error that is bounded by σ, $\|\boldsymbol{\Phi_t}(\hat{\boldsymbol{v}}_t-\boldsymbol{v}_t)\|_2<\sigma$, where $\hat{\boldsymbol{v}}_t$ is a recovered low-rank component. In the rest of the paper, we use s_0 to denote the dimensionality of the support of the source $\boldsymbol{x}_t$ and s_j to denote that of each difference vector $\boldsymbol{x}_t-\boldsymbol{z}_j$; namely, we can write $\|\boldsymbol{x}_t-\boldsymbol{z}_j\|_0=s_j$, where $j\in\{0,\ldots,J\}$. We establish the following result:

Theorem III.1. *In Problem* (23)*, let m_t be the number of measurements required to successfully separate $\boldsymbol{x}_t$ ($\|\boldsymbol{x}_t\|_0=s_0$) and $\boldsymbol{v}_t$ from compressive measurements $\boldsymbol{y}_t=\boldsymbol{\Phi}_t(\boldsymbol{x}_t+\boldsymbol{v}_t)$—where the elements of $\boldsymbol{\Phi_t}\in\mathbb{R}^{m_t\times n}$ are randomly drawn from an i.i.d. Gaussian distribution—and prior information $\boldsymbol{Z}_{t-1}:=\{\boldsymbol{z}_j\}_{j=1}^{J}$—with $\|\boldsymbol{x}_t-\boldsymbol{z}_j\|_0=s_j$ and $\boldsymbol{z}_j\in\mathbb{R}^n$—and $\boldsymbol{B}_{t-1}\in\mathbb{R}^{n\times d}$. Let λ_{m_t} denote the expected length of a zero-mean, unit-variance m_t-dimensional random Gaussian vector and $\omega(\boldsymbol{x}_t)$ the Gaussian width [44, Definition 3.1].*

1) *If $\boldsymbol{v}_t$ is not changing in time, we can recover $\boldsymbol{x}_t$ successfully with probability greater than $1-\exp(-\frac{1}{2}[\lambda_{m_t}-\omega(\boldsymbol{x}_t)]^2)$ provided that*

$$m_t\ge 2\alpha\log\Big(\frac{n}{\sum_{j=0}^{J}\beta_j s_j}\Big)+\frac{7}{5}\Big(\sum_{j=0}^{J}\beta_j s_j\Big)+1. \tag{28}$$

2) *If $\boldsymbol{v}_t$ is slowly-changing, that is, $\boldsymbol{B}_{t-1}$ and $[\boldsymbol{B}_{t-1}\ \boldsymbol{v}_t]$ are low-rank, we obtain $\hat{\boldsymbol{x}}_t$ and $\hat{\boldsymbol{v}}_t$ by solving the problem in* (23)*. Assuming that the measurement error of $\boldsymbol{v}_t$ is bounded as $\|\boldsymbol{\Phi_t}(\hat{\boldsymbol{v}}_t-\boldsymbol{v}_t)\|_2<\sigma$, we have that $\|\hat{\boldsymbol{x}}_t-\boldsymbol{x}_t\|_2\le\frac{2\sigma}{1-\sqrt{\rho}}$, where $0<\rho<1$, with probability greater than $1-\exp(-\frac{1}{2}[\lambda_{m_t}-\omega(\boldsymbol{x}_t)-(1-\sqrt{\rho})\sqrt{m_t}]^2)$ provided that*

$$m_t\ge 2\frac{\alpha}{\rho}\log\Big(\frac{n}{\sum_{j=0}^{J}\beta_j s_j}\Big)+\frac{7}{5\rho}\Big(\sum_{j=0}^{J}\beta_j s_j\Big)+\frac{3}{2\rho}, \tag{29}$$

where $\alpha=\dfrac{\epsilon^2}{\hat{\eta}^2}\sum_{j=0}^{J}\beta_j\sum_{i=1}^{s_j}w_{ji}^2$ in both (28) *and* (29)[5] *wherein $\epsilon>0$ and $\hat{\eta}=\min_j\Big\{\frac{n}{\sum_{i=1}^{n}(|x_{ti}-z_{ji}|+\epsilon)^{-1}}\Big\}$.*

Proof: The proof is given in Appendix B ■

IV. Experimental Results

We first assess the performance of RAMSIA (Section IV-A) on sparse signal recovery and the performance of CORPCA (Section IV-B) on compressive signal decomposition using synthetic data. We then assess the performance of CORPCA using real video data (Section IV-C).

A. Performance Evaluation of RAMSIA Using Synthetic Data

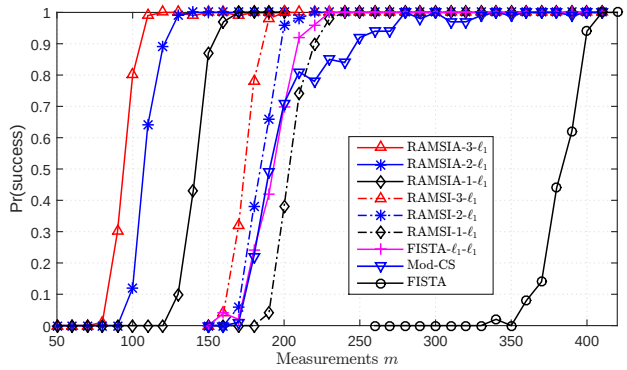
1) Experimental Setup: We consider the reconstruction of a sparse vector $\boldsymbol{x}$ given three known prior information vectors, $\boldsymbol{z}_1$, $\boldsymbol{z}_2$, $\boldsymbol{z}_3$. We generate $\boldsymbol{x}$ with $n=1000$ entries,

[5]We obtain w_{ji} and β_j from (20) and (22), respectively, given $\boldsymbol{x}_t$ and $\boldsymbol{z}_j$. Meanwhile, w_{ji} (Line 10) and β_j (Line 11) in Algorithm 2 of CORPCA are iteratively updated based on the current reconstructed $\boldsymbol{x}_t^{(k+1)}$, which is converged to $\boldsymbol{x}_t$ under the successful separation.

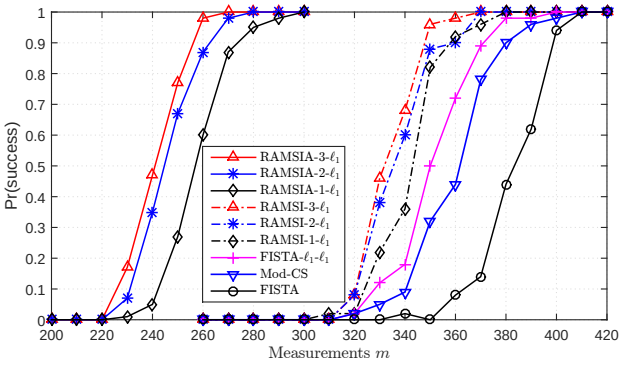
128 of which were nonzero ($\|\boldsymbol{x}\|_0 = s_0 = 128$), and which were drawn from a standard Gaussian distribution. The prior information signals were generated as $\boldsymbol{z}_j = \boldsymbol{x} - \boldsymbol{\varsigma}_j$, $j = \{1, 2, 3\}$, where $\boldsymbol{\varsigma}_j$ is a 64-sparse vector whose nonzero entries were drawn from a standard Gaussian distribution. The supports of $\boldsymbol{x}$ and $\boldsymbol{\varsigma}_j$ coincided in 51 positions. Such a prior information differed significantly from $\boldsymbol{x}$; specifically, we have $\|\boldsymbol{x}-\boldsymbol{z}_j\|_2/\|\boldsymbol{x}\|_2 \approx 0.56$. We also considered the case where the prior information signals have poorer quality, expressed by increased values for the support of $\boldsymbol{\varsigma}_j$ (specifically, $s_j = \|\boldsymbol{\varsigma}_j\|_0 = \|\boldsymbol{x} - \boldsymbol{z}_j\|_0 = 256$ and $s_j = 352$). In this case, the number of supports for which $\boldsymbol{x}$ and $\boldsymbol{\varsigma}_j$ coincide was set to 128, leading to a relative distance of $\|\boldsymbol{x} - \boldsymbol{z}_j\|_2/\|\boldsymbol{x}\|_2 \approx 1.12$ for $s_j = 256$. We clarify that we set all s_j, $j = \{1, 2, 3\}$, equal for simplicity.

2) Performance Evaluation: We compare the reconstruction performance of the proposed RAMSIA algorithm against existing methods based on ℓ_1 norm minimization. For a given number of measurements m, we compute the probability of successful recovery $\Pr(\text{success})$ as the number of times the signal $\boldsymbol{x}$ is recovered within an error $\|\widehat{\boldsymbol{x}} - \boldsymbol{x}\|_2/\|\boldsymbol{x}\|_2 \leq 10^{-2}$, divided by the total number of 100 trials (each trial considered different $\boldsymbol{x}$, $\boldsymbol{z}_1$, $\boldsymbol{z}_2$, $\boldsymbol{z}_3$, and $\boldsymbol{\Phi}$).

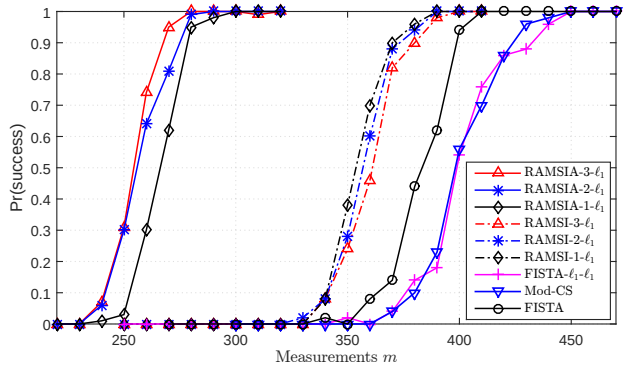
Let RAMSIA-1-ℓ_1, RAMSIA-2-ℓ_1, RAMSIA-3-ℓ_1 denote the proposed RAMSIA algorithm configured to use one (a.k.a., $\boldsymbol{z}_1$), two (a.k.a., $\boldsymbol{z}_1$, $\boldsymbol{z}_2$), and three (a.k.a., $\boldsymbol{z}_1$, $\boldsymbol{z}_2$, and $\boldsymbol{z}_3$) prior information signals, respectively. We have set $\lambda = 10^{-5}$ and $\epsilon = 0.8$ in RAMSIA, as experimentation revealed that these values led to the best performance. Similarly, RAMSI-3-ℓ_1, RAMSI-2-ℓ_1, RAMSI-1-ℓ_1 are the corresponding three versions of our previous algorithm in [48]. Moreover, let FISTA and FISTA-ℓ_1-ℓ_1 denote the versions of the FISTA [45] algorithm solving the ℓ_1 and the ℓ_1-ℓ_1 minimization problem [41], [42], respectively (where the latter is using the signal $\boldsymbol{z}_1$ as prior information). Finally, Mod-CS corresponds to the Modified-CS [52] algorithm. We assess the recovery performance of the aforementioned algorithms via reporting the success rate versus the number of measurements m in Fig. 2. Specifically, Fig. 2(a) and Figs. 2(b), (c) present the results for the experiments corresponding to the good ($s_j = 64$) and the poor ($s_j = 256$ and $s_j = 352$) quality prior information, respectively. It is clear that RAMSIA-3-ℓ_1 successfully leverages the correlation of the various prior information signals with the target signal, delivering the highest probability of success in comparison to all competing algorithms for all experiments. It is also evident that the performance of RAMSIA increases with the number of used prior signals. As anticipated, the increase in performance is more apparent when the quality of the prior information is high [compare, for example, the results in Fig. 2(a) and Fig. 2(c)]. Importantly, RAMSIA-1-ℓ_1 outperforms significantly (and systematically) RAMSI-1-ℓ_1 [48], FISTA-ℓ_1-ℓ_1 [41], [42], and Mod-CS [52], which also consider only one prior information signal aiding the reconstruction. Interestingly, Fig. 2(c) reveals that the accuracy of FISTA-ℓ_1-ℓ_1 and Mod-CS becomes lower than that of the conventional FISTA algorithm; namely, in these algorithms the poor-quality prior information hinders the recovery performance. This does not happen for RAMSIA-1-ℓ_1, which always improves over FISTA irrespective of the side information quality. These results highlight the advantages of the proposed algorithm compared to the state of the art.



(a) $s_j = 64$



(b) $s_j = 256$



(c) $s_j = 352$

Fig. 2. Probability of success in recovering the synthetic signal $\boldsymbol{x} \in \mathbb{R}^{1000}$ (with sparsity $s_0 = 128$) versus the number of measurements m. The reconstruction is aided by a number of prior information signals ($\boldsymbol{z}_j, j = \{1, 2, 3\}$) with sparsity of the difference vector $\boldsymbol{\varsigma}_j = \boldsymbol{x} - \boldsymbol{z}_j$ equal to: (a) $s_j = 64$, (b) $s_j = 256$, and (c) $s_j = 352$, $j = \{1, 2, 3\}$.

3) Prior-Information Quality Analysis: We here analyze further how the prior information quality influences the number of measurements required to successfully reconstruct the signal. Specifically, we keep the sparsity of $\boldsymbol{x}$ constant at $s_0 = 128$ and assess the performance of RAMSIA, our previous RAMSI in [48], and the other algorithms when s_j—that is, the sparsity of $\boldsymbol{\varsigma}_j$, $j = \{1, 2, 3\}$—varies from 20 to 400 (for simplicity, we set $s_1 = s_2 = s_3$). In this experiment, we assume successful recovery when $\Pr(\text{success}) \geq 0.98$. Fig. 3 depicts the number of measurements required for successful reconstruction versus the value of s_j, $j = \{1, 2, 3\}$ for the different algorithms. Evidently, using prior information reduces

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIP.2018.2831915, IEEE Transactions on Image Processing

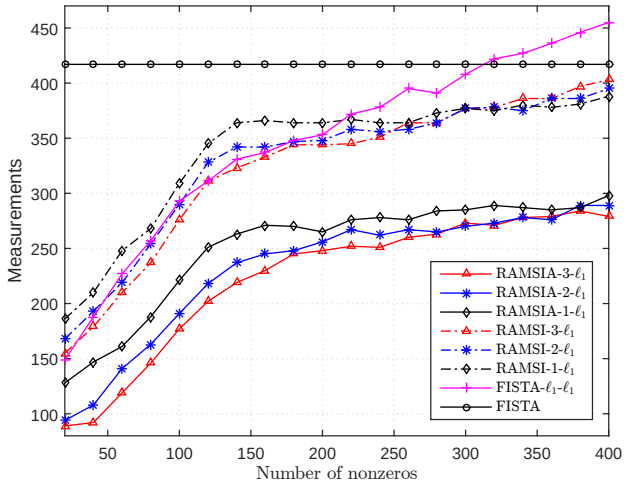


Fig. 3. The number of measurements required to recover the synthetic signal $\boldsymbol{x} \in \mathbb{R}^{1000}$ (with sparsity $s_0 = 128$) versus the sparsity $s_1 = s_2 = s_3$ of the difference vectors $\boldsymbol{\varsigma}_j = \boldsymbol{x} - \boldsymbol{z}_j$, $j = \{1, 2, 3\}$.

the number of measurements compared to FISTA. However, at $s_j = 315$, the Modified-CS and FISTA-ℓ_1-ℓ_1 performance become worse than that of FISTA, thereby illustrating the limitation of the ℓ_1-ℓ_1 minimization approach. On the contrary, the proposed weighted n-ℓ_1 minimization strategy leads to systematically improved performance compared to FISTA.

B. Performance Evaluation of CORPCA Using Synthetic Data

We evaluate the performance of Algorithm 2 employing the proposed n-ℓ_1 minimization method as well as the existing ℓ_1 [45] and ℓ_1-ℓ_1 [41], [42] minimization methods for the sparse recovery step. We denote the three configurations as CORPCA-n-ℓ_1, CORPCA-ℓ_1-ℓ_1, and CORPCA-ℓ_1, respectively. We also compare CORPCA against RPCA [5], GRASTA [18], and ReProCS [37]. RPCA [5] is a batch-based method assuming full access to the data, while GRASTA [18] and ReProCS [37] are online methods. For compressive measurements, ReProCS (see Algorithm 4 in [37]) recovers the sparse components, while GRASTA recovers the low-rank components (GRASTA was assessed using full video data for the separation, see the experiments in [18]).

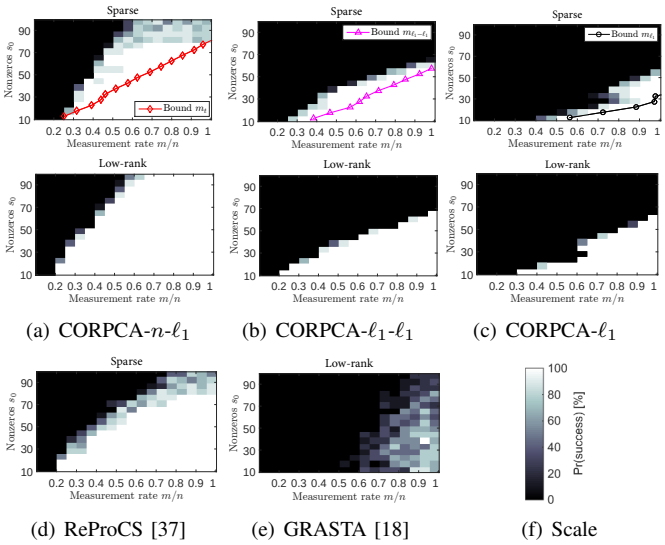
We generate our data as follows. First, we generate the low-rank component as $\boldsymbol{L} = \boldsymbol{U}\boldsymbol{V}^{\mathrm{T}}$, where $\boldsymbol{U} \in \mathbb{R}^{n \times r}$ and $\boldsymbol{V} \in \mathbb{R}^{(d+q) \times r}$ are random matrices whose entries are drawn from the standard normal distribution. We set $n = 500$, $r = 5$ (the rank of $\boldsymbol{L}$), and $d = 100$ the number of vectors for training and $q = 100$ the number of testing vectors; this yields $\boldsymbol{L} = [\boldsymbol{v}_1 \dots \boldsymbol{v}_{d+q}]$. Secondly, we generate $\boldsymbol{X} = [\boldsymbol{x}_1 \dots \boldsymbol{x}_{d+q}]$. Specifically, at time instance $t = 1$, we draw $\boldsymbol{x}_1 \in \mathbb{R}^n$ from the standard normal distribution with s_0 nonzero elements. Then, we generate a sequence of correlated sparse vectors $\boldsymbol{x}_t$, $t = \{2, 3, \dots, d+q\}$, where each $\boldsymbol{x}_t$ satisfies $\|\boldsymbol{x}_t - \boldsymbol{x}_{t-1}\|_0 = s_0/2$. As this could lead to $\|\boldsymbol{x}_t\|_0 > s_0$, we add the constraint $\|\boldsymbol{x}_t\|_0 \in [s_0, s_0 + 15]$. Whenever $\|\boldsymbol{x}_t\|_0 > s_0 + 15$, we reset $\boldsymbol{x}_t$ to $\|\boldsymbol{x}_t\|_0 = s_0$ by setting $\|\boldsymbol{x}_t\|_0 - s_0$ randomly selected positions to zero. Thirdly, we initialize the prior information; in order to address real scenarios, where we do not know the sparse and low-rank components, we use the batch-based RPCA [5] method to separate the training set $\boldsymbol{M}_0 = [\boldsymbol{x}_1 + \boldsymbol{v}_1 \ \dots \ \boldsymbol{x}_d + \boldsymbol{v}_d]$ so as to obtain $\boldsymbol{B}_0 = [\boldsymbol{v}_1 \ \dots \ \boldsymbol{v}_d]$. In this experiment, we use three (a.k.a., $J = 3$) sparse components as prior information and we set $\boldsymbol{Z}_0 := \{\boldsymbol{0}, \boldsymbol{0}, \boldsymbol{0}\}$.

We then evaluate the CORPCA method (see Section III-B) on the test set of vectors $\boldsymbol{M} = [\boldsymbol{x}_{d+1} + \boldsymbol{v}_{d+1} \ \dots \ \boldsymbol{x}_{d+q} + \boldsymbol{v}_{d+q}]$. Specifically, we vary s_0 (from 10 to 90) and the measurement rate m/n, and we assess the probability of successful reconstruction $\Pr(\text{success})$. In this experiment we attempt to recover both the sparse $\boldsymbol{x}_t$ and the low-rank $\boldsymbol{v}_t$ component for all $t = d+1, d+2, \dots, d+q$. Hence, we assess the probability of success for the sparse $\Pr_{\text{sparse}}(\text{success})$ and the low-rank $\Pr_{\text{low-rank}}(\text{success})$ component, averaged over the test vectors $t = d+1, d+2, \dots, d+q$. $\Pr_{\text{sparse}}(\text{success})$ [resp. $\Pr_{\text{low-rank}}(\text{success})$] is defined as the number of times in which the sparse component $\boldsymbol{x}_t$ (resp. the low-rank component $\boldsymbol{v}_t$) is recovered within an error $\|\widehat{\boldsymbol{x}}_t - \boldsymbol{x}_t\|_2 / \|\boldsymbol{x}_t\|_2 \leq 10^{-2}$ (resp. $\|\widehat{\boldsymbol{v}}_t - \boldsymbol{v}_t\|_2 / \|\boldsymbol{v}_t\|_2 \leq 10^{-2}$) divided by the total 50 Monte Carlo simulations. We note that in Algorithm 2 we have set $\epsilon = 0.8$, $\lambda = 1/\sqrt{n}$, and $\mu = 10^{-3}$. In addition, we evaluate our bound for the n-ℓ_1 minimization method in (29) as well as the corresponding bounds for ℓ_1 and ℓ_1-ℓ_1 minimization[6]. We have set the parameter ρ to $0.8/3$, $0.6/3$, $0.4/3$ for the bounds of n-ℓ_1, ℓ_1-ℓ_1, and ℓ_1 minimization, respectively, as experimentation has shown that these values achieve the most accurate prediction of the real performance.

The results in Fig. 4 demonstrate the efficiency of the proposed CORPCA algorithm employing the proposed n-ℓ_1 minimization method. At specific sparsity levels, we can recover the 500-dimensional data from measurements of much lower dimensions [$m/n = 0.3$ to 0.6, see the white areas in Fig. 4(a)]. It is also clear that the ℓ_1 and ℓ_1-ℓ_1 minimization methods [see Figs. 4(b), 4(c)] lead to a higher number of measurements, thereby illustrating the benefit of incorporating multiple side information into the problem. As mentioned before, the ReProCS and GRASTA methods focus primarily on recovering the sparse components and the low-rank components from compressive measurements, respectively. Fig. 4(d) shows that the performance of ReProCS is worse than that of CORPCA-n-ℓ_1 [see Fig. 4(a)]. Moreover, Fig. 4(e) shows that GRASTA delivers lower low-rank recovery performance than the proposed CORPCA method. It is also worth observing that the measurement bound for recovering the sparse component by CORPCA [red line in Fig. 4(a)] is sharper than the existing bounds for ℓ_1-ℓ_1 and ℓ_1 minimization, depicted in Fig. 4(b), and Fig. 4(c), respectively.

Moreover, we evaluate CORPCA given that only the subsampled data of the training set $\boldsymbol{M}_0$ is available. To obtain the prior $\boldsymbol{B}_0$, we perform compressive batch RPCA—using the SpaRCS [23] method—on the subsampled set of $\boldsymbol{M}_0$ via a projection $\boldsymbol{\Phi} \in \mathbb{R}^{m \times n}$. The testing set $\boldsymbol{M}$ is generated similarly with the above full training set. We vary s_0 from 5 to 90 and set $n = 512$, $r = 5$ (the rank of $\boldsymbol{L}$), the number of training vectors $d = 128$, and the number of

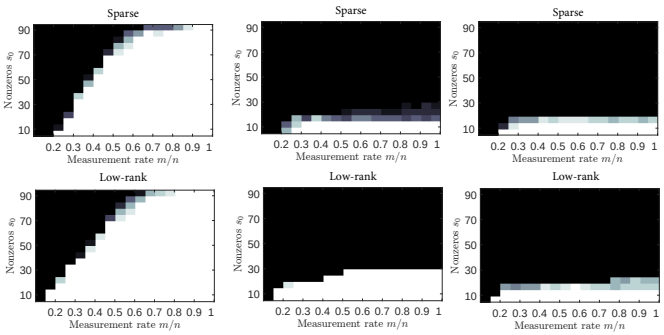
[6]The bound for the number of measurements required by ℓ_1 and ℓ_1-ℓ_1 minimization to achieve successful recovery of the sparse component assuming that the low-rank component is slowly-varying (see Theorem III.1) is given by $m_{\ell_1} \geq 2\frac{s_0}{\rho}\log\frac{n}{s_0} + \frac{7}{5}\frac{s_0}{\rho} + \frac{3}{2\rho}$ [44], and $m_{\ell_1\text{-}\ell_1} \geq 2\frac{h}{\rho}\log\left(\frac{n}{s_0+\xi/2}\right) + \frac{7}{5\rho}\frac{s_0}{\rho}\left(s_0 + \frac{\xi}{2}\right) + \frac{3}{2\rho}$ [41], [42], respectively.



(a) CORPCA-n-ℓ_1 (b) CORPCA-ℓ_1-ℓ_1 (c) CORPCA-ℓ_1

(d) ReProCS [37] (e) GRASTA [18] (f) Scale

Fig. 4. Average success probabilities for CORPCA (for recovering $\boldsymbol{x}_t, \boldsymbol{v}_t \in \mathbb{R}^{500}$), ReProCS [37] (for $\boldsymbol{x}_t$), and GRASTA [18] (for $\boldsymbol{v}_t$).



(a) Full (b) 70% (c) 35%

Fig. 5. Average success probabilities for CORPCA (for recovering $\boldsymbol{x}_t, \boldsymbol{v}_t \in \mathbb{R}^{512}$) using (a) a full training set and subsampled training sets, (b) 70% and (c) 35%.

testing vectors $q = 128$ (following that the SpaRCS code[7] requires dimensions that are a power of 2). The corresponding results are shown in Fig. 5, where $m/n = \{1,\ 0.7,\ 0.35\}$ are tested. We observe that the performance of COPRCA under the subsampled training sets [see Figs. 5(b) and 5(c)] is significantly degraded. These degradations are caused by the poor quality $\boldsymbol{B}_0$, obtained by SpaRCS. In these experiments, SpaRCS does not work well for a given rank $r = 5$ of $\boldsymbol{L}$ and higher supports $s_0 > 15$ on the subsampled training set. In realistic scenarios, we should take the characteristics of the rank of low-rank components and the sparse degree of sparse components into account to avoid obtaining poor $\boldsymbol{B}_0$ from the subsampled training set.

C. Compressive Video Foreground-Background Separation

We now assess our CORPCA method in the application of compressive video background-foreground separation using real video content and compare it against existing methods, namely, GRASTA [18] and ReProCS [37].

1) Visual Evaluation: We consider several video sequences [47], namely, `Bootstrap` (rescaled to 60×80 pixels), `Curtain` (rescaled to 64×80 pixels), `Hall` (rescaled to 72×88 pixels) and `Fountain` (rescaled to 64×80 pixels), where `Bootstrap` and `Hall` have static backgrounds and `Curtain` and `Fountain` have dynamic backgrounds (see Fig. 6). For all methods, we use the first $d = 100$ frames for training and the subsequent $q = 2955$ (`Bootstrap`), $q = 2864$ (`Curtain`), $q = 3484$ (`Hall`), and $q = 423$ (`Fountain`) frames for evaluation. We use three sparse components as prior information: $\hat{\boldsymbol{x}}_{t-1}$, $\hat{\boldsymbol{x}}_{t-2}$, and $\hat{\boldsymbol{x}}_{t-3}$. We consider the immediately previously reconstructed foreground as the first side information signal, i.e., $\boldsymbol{z}_1 = \hat{\boldsymbol{x}}_{t-1}$. The other two side information signals are formed by applying motion-compensated extrapolation using the three previously reconstructed frames. Specifically, we perform forward optical-flow-based [50], [53] motion estimation from $\hat{\boldsymbol{x}}_{t-2}$ to $\hat{\boldsymbol{x}}_{t-1}$ (resp., $\hat{\boldsymbol{x}}_{t-3}$ to $\hat{\boldsymbol{x}}_{t-1}$) and then apply the motion vectors on $\hat{\boldsymbol{x}}_{t-1}$ to generate $\boldsymbol{z}_2$ (resp. $\boldsymbol{z}_3$).

We first consider the case where we have full access to the video frames (that is, the number of measurements m is equal to the dimension n of the data); the visual results of the various methods are illustrated in Fig. 6. For both video sequences, CORPCA delivers superior visual results than the other methods, which suffer from less details in the separated foreground and noise in the background. Fig. 7 presents the results of CORPCA under various compressive rates m/n. The results show that we can recover the foreground and background even by accessing a small number of measurements; for instance, with $m/n = 0.6$ and for `Bootstrap` [Fig. 7(a)], `Hall` [Fig. 7(c)] and $m/n = 0.4$ for `Curtain` [Fig. 7(b)], `Fountain` [Fig. 7(d)]. `Bootstrap` and `Hall` requires more measurements than `Curtain` and `Fountain` due to the more complex foreground. For comparison, we also illustrate the visual results of the foreground recovered by ReProCS in Fig. 8. It is clear that the foreground images recovered with ReProCS have a poorer visual quality compared to CORPCA even at a high rate $m/n = 0.8$. The original and the reconstructed video sequences are available online [51].

In addition, we illustrate the results of CORPCA by directly using the reconstructed foregrounds without motion compensation as well as considering a different number (J) of priors. Fig. 9 shows the visual results of CORPCA for `Bootstrap` (frame no. 2213) and `Curtain` (frame no. 2866) using three priors, $\boldsymbol{z}_j = \widehat{\boldsymbol{x}}_{t-j}$ ($1 \le j \le 3$) taken directly from previously reconstructed foregrounds. These results are worse than those of CORPCA with motion compensation (see Fig. 7); particularly, at rates $m/n = 0.4$ and $m/n = 0.2$, Figs. 9(a) and 9(b) show poorer foregrounds than the ones depicted in Figs. 7(a) and 7(b), respectively. To demonstrate the influence of the number (J) of foreground priors, Fig. 10 shows the visual results of CORPCA using one and five priors. Setting $J = 1$ [Fig. 10(a)] delivers worse results than setting $J = 5$ [Fig. 10(a)] and $J = 3$ [Fig. 7(a)]. Furthermore, the results using $J = 5$ are only slightly better than those using $J = 3$, concluding that using a reasonable number of priors $J = 3$ is sufficient to capture the temporal correlation.

2) Quantitative Results: We now present quantitative results using the *receiver operating curve* (ROC) metric [54] (the *true positives* and *false positives* metrics are defined as in [54]). The ROC results for CORPCA deploying differ-

[7]The code is available at https://www.ece.rice.edu/ aew2/sparcs.html

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIP.2018.2831915, IEEE Transactions on Image Processing

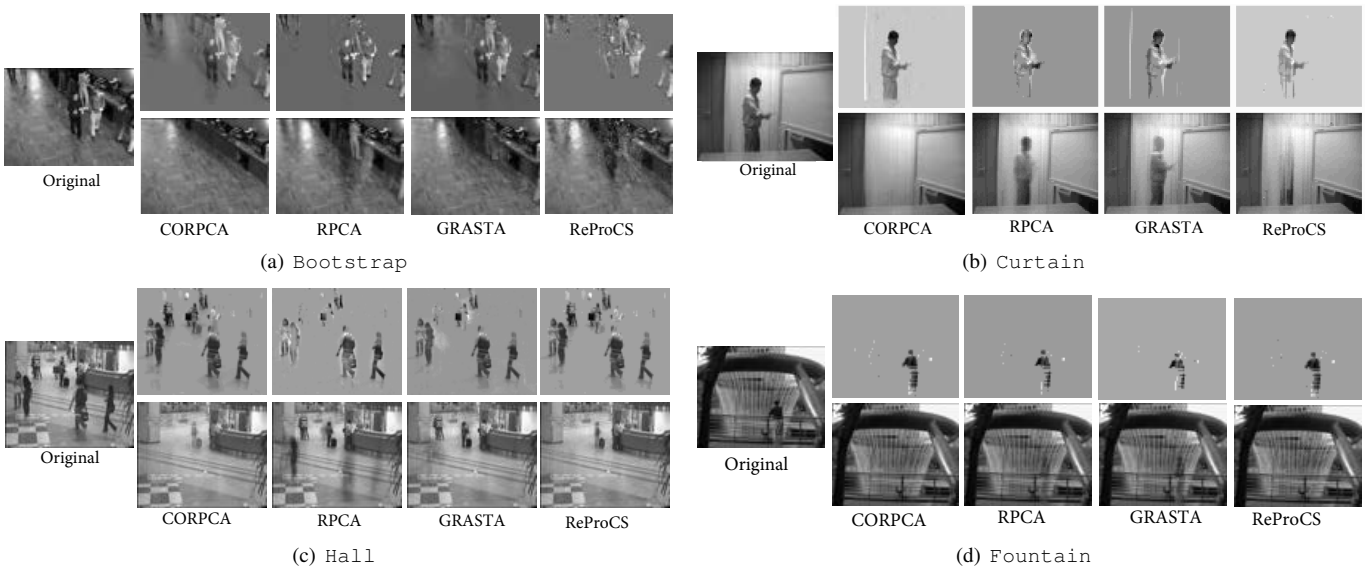


Fig. 6. Visual results when assuming access to the entire data vectors: (a) Bootstrap (frame no. 2213), (b) Curtain (frame no. 2866), (c) Hall (frame no. 1963), and (d) Fountain (frame no. 480).

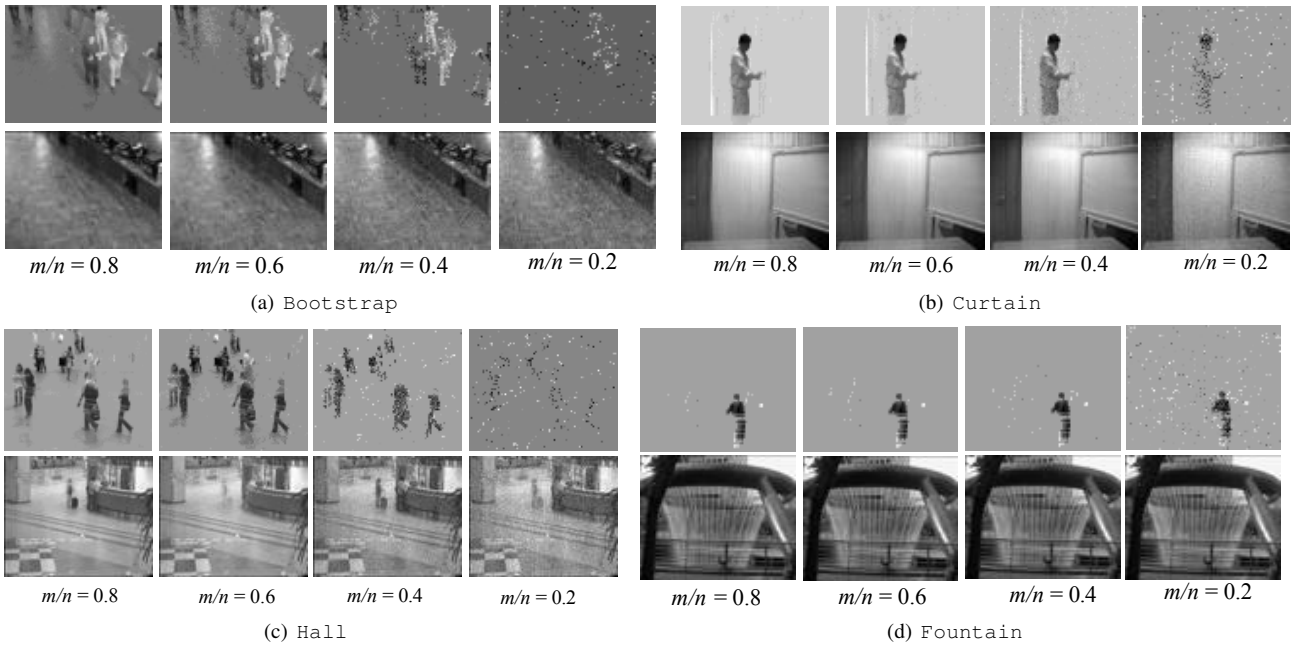
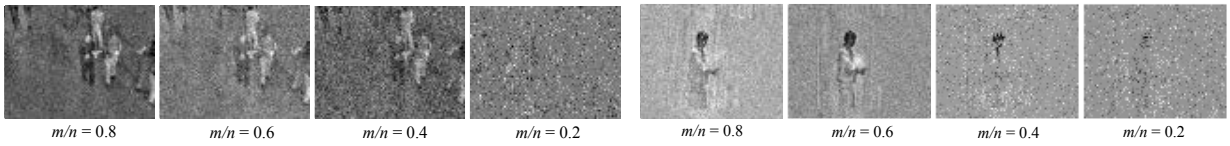


Fig. 7. Visual performance of CORPCA under different measurement rates: (a) Bootstrap (frame no. 2213), (b) Curtain (frame no. 2866), (c) Hall (frame no. 1963), and (d) Fountain (frame no. 480).

ent recovery algorithms, namely n-ℓ_1, ℓ_1-ℓ_1, and ℓ_1 minimization, are depicted in Fig. 11 for the Bootstrap and Curtain sequences. It is clear that CORPCA-ℓ_1-ℓ_1 [Figs. 11(b), 11(e)] and CORPCA-ℓ_1 [Figs. 11(c), 11(f)] deliver worse performance than CORPCA-n-ℓ_1 [Figs. 11(a), 11(d)]. It is worth observing that CORPCA-n-ℓ_1 delivers high performance even at low rates, that is, at $m/n = 0.6$ for Bootstrap [Fig. 11(a)], and $m/n = 0.4$ for Curtain [see Fig. 11(d)]. Fig. 12 illustrates the ROCs of CORPCA, RPCA, GRASTA, and ReProCS when assuming full data access (i.e., $m/n = 1$). The results show that CORPCA delivers higher performance than the other methods, especially for the Curtain video sequence [see Fig. 12(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. 13 show that CORPCA achieves a relatively high performance with a small number of measurements. In addition, the ROC results for ReProCS are quickly degrading even at a high compressive measurement rate $m/n = 0.8$.

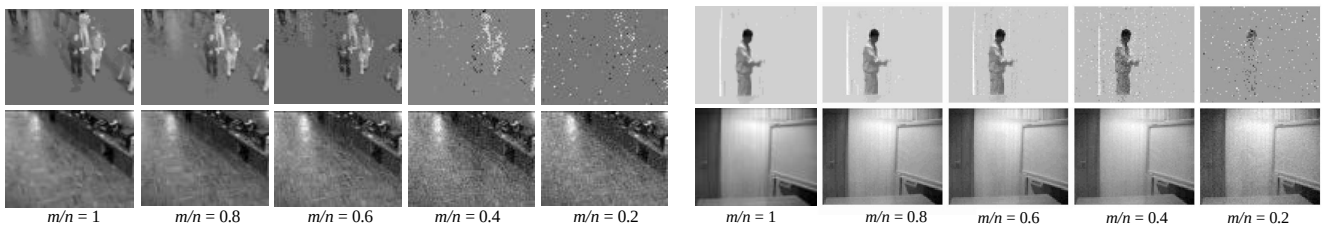
Memory and Computational Needs: In the considered video analysis application, batch RPCA [5] needs to load all raw video samples—of size $n \times p$ pixels (where n is the number of pixels per frame and p is the number of frames)—in memory to perform the decomposition. Batch-based compressed SpaRCS [23] needs to load a reduced amount of video samples, namely,

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIP.2018.2831915, IEEE Transactions on Image Processing



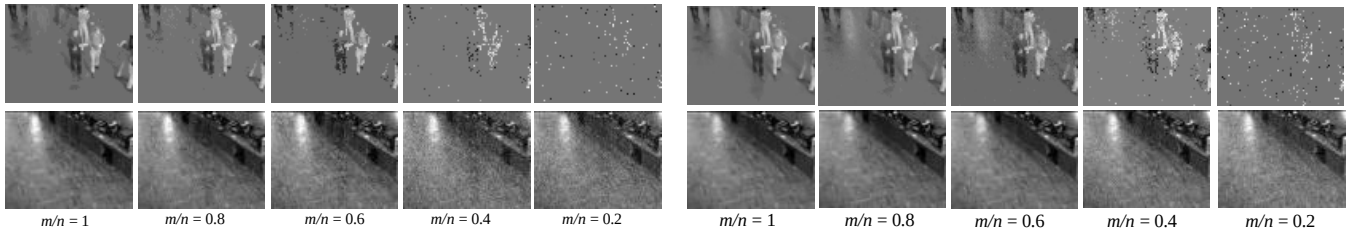
(a) `Bootstrap` (b) `Curtain`

Fig. 8. Foreground recovered using ReProCS [37] under different measurement rates: (a) `Bootstrap` (frame no. 2213) and (b) `Curtain` (frame no. 2866).



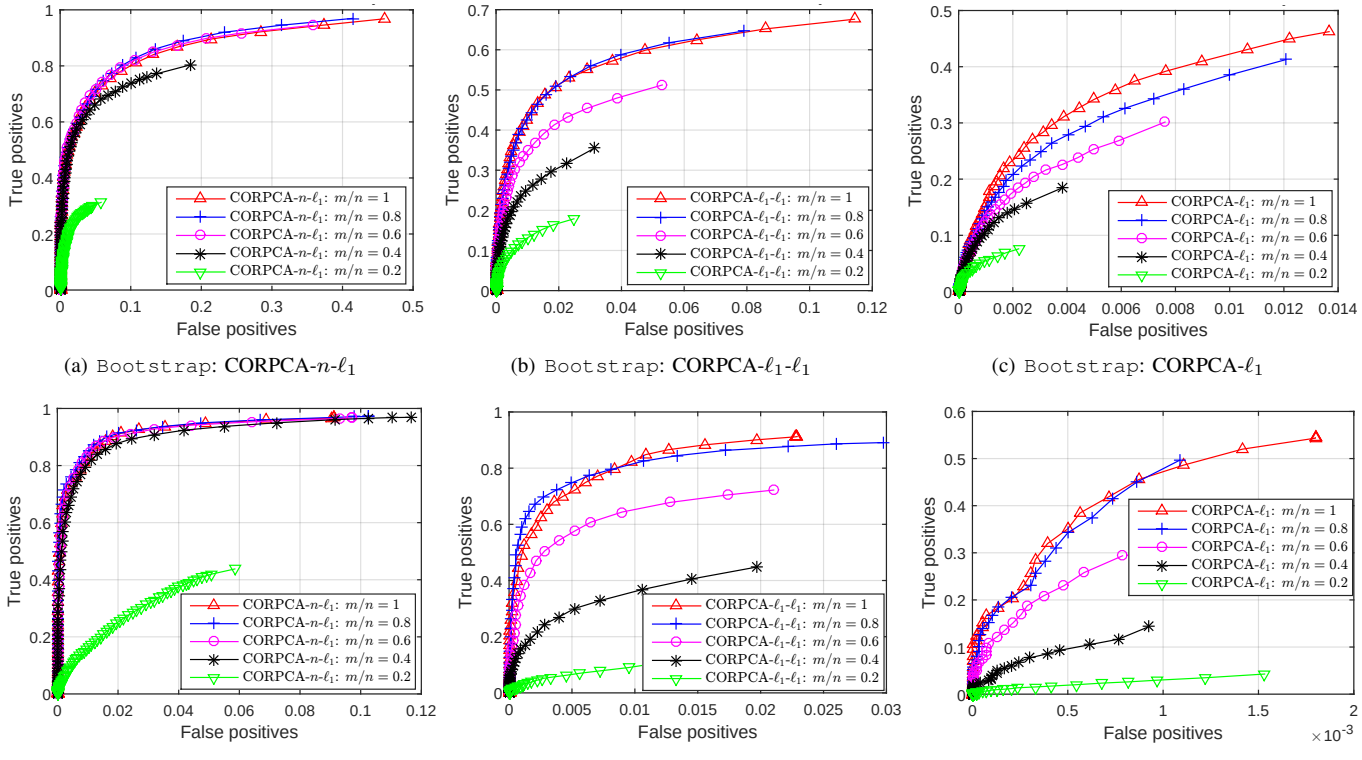
(a) `Bootstrap` (b) `Curtain`

Fig. 9. Visual performance of CORPCA for (a) `Bootstrap` (frame no. 2213) and (b) `Curtain` (frame no. 2866) using three ($J = 3$) directly reconstructed foregrounds as priors without motion compensation.



(a) $J = 1$ (b) $J = 5$

Fig. 10. Visual performance of CORPCA for `Bootstrap` (frame no. 2213) using different numbers of foreground priors: (a) $J = 1$ and (b) $J = 5$.

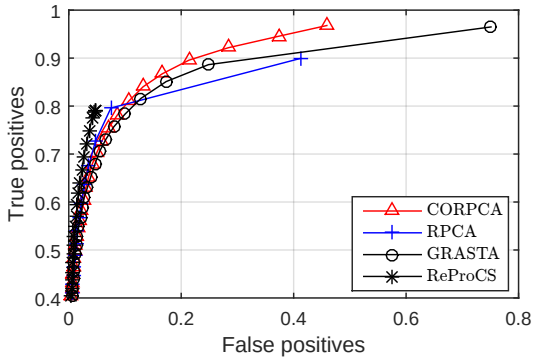


(a) `Bootstrap`: CORPCA-n-ℓ_1 (b) `Bootstrap`: CORPCA-ℓ_1-ℓ_1 (c) `Bootstrap`: CORPCA-ℓ_1

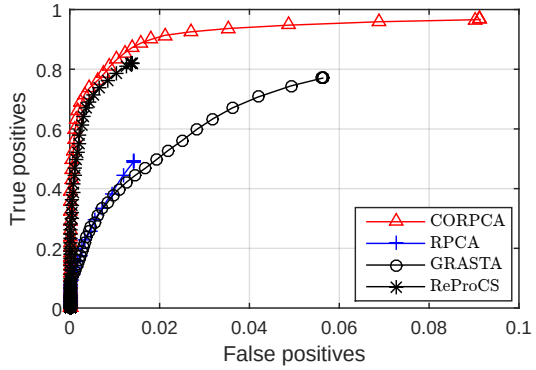
(d) `Curtain`: CORPCA-n-ℓ_1 (e) `Curtain`: CORPCA-ℓ_1-ℓ_1 (f) `Curtain`: CORPCA-ℓ_1

Fig. 11. ROC for CORPCA deploying different sparse recovery algorithms with different measurement rates.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIP.2018.2831915, IEEE Transactions on Image Processing

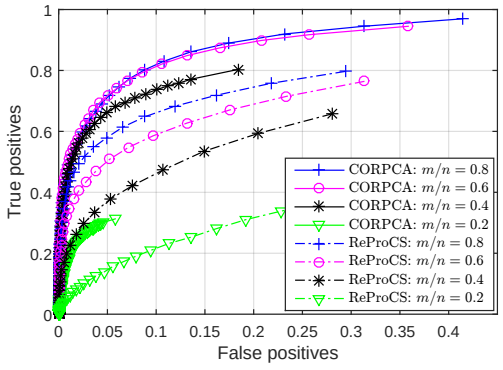


(a) Bootstrap

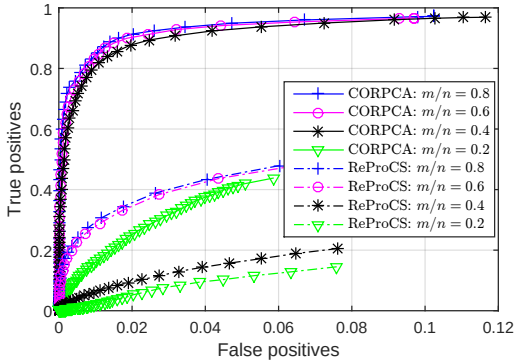


(b) Curtain

Fig. 12. ROC for the different separation methods with full data $m/n = 1$.



(a) Bootstrap



(b) Curtain

Fig. 13. ROC for CORPCA and ReProCS [37] with compressive measurements, $m/n = \{0.8, 0.6, 0.4, 0.2\}$.

TABLE I
COMPUTATIONAL TIMES PER FRAME FOR BOOTSTRAP AND CURTAIN

		CORPCA	GRASTA [18]	ReProCS [37]
Bootstrap (sec/frame)	$m/n = 1$	0.75	0.003	0.29
	$m/n = 0.6$	1.72		1.39
Curtain (sec/frame)	$m/n = 1$	0.22	0.003	0.13
	$m/n = 0.4$	1.59		0.82

$m \times p$ pixels, where $m \ll n$ is the numbered of measurements. Both methods have high computational and memory usage demands, especially when the number of processed video frames increases. Alternatively, recursive methods like GRASTA [18], ReProCS [37], and CORPCA only load one video frame at each time instance and do not have a processing delay of p frames. The advantage of CORPCA over non-compressed online RPCA—such as incremental PCP [17]—is that it access subsampled data per frame (namely, $m \ll n$ instead of n samples), thereby reducing further the associated data storage and communication requirements.

We evaluate the computational time of CORPCA by comparing it against other online methods, that is, GRASTA [18] and ReProCS [37]. Matlab implementations of these algorithms are executed on a 64-bit desktop with a 3.5GHz Core i5-CPU and 8GB of memory, running Windows 7, and the execution times are measured. Table I reports the average time per frame for processing full data ($m/n = 1$) and subsampled data—with $m/n = 0.6$ and $m/n = 0.4$, respectively—from the Bootstrap and Curtain video sequences. The results show that CORPCA requires the highest computational time compared to the other methods. This is because (*i*) CORPCA requires additional computations for incorporating multiple priors in the optimization problem and (*ii*) the code is not optimized for speed. By using a C++ implementation running on GPUs, CORPCA can be accelerated to process real-time high-resolution videos.

V. CONCLUSION

This paper proposed a compressive online robust PCA (CORPCA) algorithm that decomposes a data vector per time instance into a low-rank and a sparse component using low-dimensional measurements. A key component of CORPCA is that it incorporates multiple prior information in the sparse recovery problem via establishing and solving an n-ℓ_1 minimization problem. Moreover, we established the theoretical bounds on the number of measurements to guarantee successful recovery by the n-ℓ_1 minimization problem. Numerical results have shown the efficiency of CORPCA and the consistency with the theoretical bounds. We have also assessed our method against the state of the art in compressive video foreground-background separation. The results revealed the advantage of incorporating multiple prior information in the problem and demonstrated the superior performance improvement offered by CORPCA compared to the existing methods.

VI. ACKNOWLEDGEMENT

This work was supported by the Humboldt Research Fellowship, the Alexander von Humboldt Foundation, 53173 Bonn, Germany.

APPENDIX A
THE PROXIMAL OPERATOR FOR RAMSIA

Proposition A.1. *The proximal operator* $\Gamma_{\frac{1}{L}g}(\boldsymbol{x})$ *in* (7) *for the problem of signal recovery with multiple side information, for*

which $g(\boldsymbol{x}) = \lambda \sum_{j=0}^{J} \beta_j \|\mathbf{W}_j(\boldsymbol{x}-\boldsymbol{z}_j)\|_1$, *is given by*

$$\Gamma_{\frac{1}{L}g}(x_i)=\begin{cases} x_i-\frac{\lambda}{L}\sum_{j=0}^{J}\beta_j w_{ji}(-1)^{\mathfrak{b}(l<j)} & \text{if (17a);} \\ z_{li} & \text{if (17b);} \end{cases} \tag{30}$$

Proof: We compute the proximal operator $\Gamma_{\frac{1}{L}g}(\boldsymbol{x})$ (7) with $g(\boldsymbol{x}) = \lambda\sum_{j=0}^{J}\beta_j||\mathbf{W}_j(\boldsymbol{x}-\boldsymbol{z}_j)||_1$. From (7), $\Gamma_{\frac{1}{L}g}(\boldsymbol{x})$ is expressed by:

$$\Gamma_{\frac{1}{L}g}(\boldsymbol{x})=\arg\min_{\boldsymbol{v}\in\mathbb{R}^n}\left\{\frac{\lambda}{L}\sum_{j=0}^{J}\beta_j||\mathbf{W}_j(\boldsymbol{v}-\boldsymbol{z}_j)||_1+\frac{1}{2}||\boldsymbol{v}-\boldsymbol{x}||_2^2\right\}. \tag{31}$$

We note that both terms in (31) are separable in $\boldsymbol{v}$ and thus we can minimize each element v_i of $\boldsymbol{v}$ individually as

$$\Gamma_{\frac{1}{L}g}(x_i)=\arg\min_{v_i\in\mathbb{R}}\left\{h(v_i)=\frac{\lambda}{L}\sum_{j=0}^{J}\beta_j w_{ji}|v_i-z_{ji}|+\frac{1}{2}(v_i-x_i)^2\right\}. \tag{32}$$

We consider the $\partial h(v_i)/\partial v_i$. Without loss of generality, we assume $-\infty \le z_{0i} \le z_{1i} \le ... \le z_{Ji} \le \infty$. For convenience, let us denote $z_{(-1)i} = -\infty$ and $z_{(J+1)i} = \infty$. When v_i is located in one of the intervals, we suppose $v_i \in (z_{li}, z_{(l+1)i})$ with $-1 \le l \le J$, where $\partial h(v_i)$ exists. Taking the derivative of $h(v_i)$ in $(z_{li}, z_{(l+1)i})$ delivers

$$\frac{\partial h(v_i)}{\partial v_i}=\frac{\lambda}{L}\sum_{j=0}^{J}\beta_j w_{ji}\text{sign}(v_i-z_{ji})+(v_i-x_i), \tag{33}$$

where $\text{sign}(.)$ is a sign function. From (18) $\mathfrak{b}(.)$ denotes a boolean function, consequently, $\text{sign}(v_i-z_{ji})=(-1)^{\mathfrak{b}(l<j)}$ and using (33) we rewrite:

$$\frac{\partial h(v_i)}{\partial v_i}=\frac{\lambda}{L}\sum_{j=0}^{J}\beta_j w_{ji}(-1)^{\mathfrak{b}(l<j)}+(v_i-x_i). \tag{34}$$

When setting $\partial h(v_i)/\partial v_i = 0$ to minimize $h(v_i)$, we derive:

$$v_i=x_i-\frac{\lambda}{L}\sum_{j=0}^{J}\beta_j w_{ji}(-1)^{\mathfrak{b}(l<j)}. \tag{35}$$

The v_i in (35) is only valid in $(z_{li}, z_{(l+1)i})$, i.e.,

$$z_{li}+\frac{\lambda}{L}\sum_{j=0}^{J}\beta_j w_{ji}(-1)^{\mathfrak{b}(l<j)}<x_i<z_{(l+1)i}+\frac{\lambda}{L}\sum_{j=0}^{J}\beta_j w_{ji}(-1)^{\mathfrak{b}(l<j)}. \tag{36}$$

In the remaining range value of x_i, namely, in the case that

$$z_{li}+\frac{\lambda}{L}\sum_{j=0}^{J}\beta_j w_{ji}(-1)^{\mathfrak{b}(l-1<j)}\le x_i\le z_{li}+\frac{\lambda}{L}\sum_{j=0}^{J}\beta_j w_{ji}(-1)^{\mathfrak{b}(l<j)}, \tag{37}$$

we prove that the minimum of $h(v_i)$ (32) is obtained when $v_i = z_{li}$ in the following Lemma A.2.

Lemma A.2. *Given x_i belonging to the intervals represented in* (37)*, $h(v_i)$ in* (32) *is minimum when $v_i = z_{li}$.*

Proof: We re-express $h(v_i)$ in (32) as

$$h(v_i)=\frac{\lambda}{L}\sum_{j=0}^{J}\beta_j w_{ji}\Big|(v_i-z_{li})-(z_{ji}-z_{li})\Big|+\frac{1}{2}\Big((v_i-z_{li})-(x_i-z_{li})\Big)^2. \tag{38}$$

Applying the inequality $|a-b| \ge |a|-|b|$, where $a, b \in \mathbb{R}$, to the first term and expanding the second term in (38), we obtain:

$$\begin{aligned} h(v_i)\ge\ & \frac{\lambda}{L}\sum_{j=0}^{J}\beta_j w_{ji}|v_i-z_{li}|-\frac{\lambda}{L}\sum_{j=0}^{J}\beta_j w_{ji}|z_{ji}-z_{li}|+ \\ & \frac{1}{2}(v_i-z_{li})^2-(v_i-z_{li})(x_i-z_{li})+\frac{1}{2}(x_i-z_{li})^2. \end{aligned} \tag{39}$$

It can be noted that $-(v_i-z_{li})(x_i-z_{li}) \ge -|v_i-z_{li}||x_i-z_{li}|$. Thus the inequality in (39) is equivalent to:

$$\begin{aligned} h(v_i)\ge\ & |v_i-z_{li}|\frac{\lambda}{L}\sum_{j=0}^{J}\beta_j w_{ji}-|v_i-z_{li}||x_i-z_{li}| \\ & +\frac{1}{2}(v_i-z_{li})^2-\frac{\lambda}{L}\sum_{j=0}^{J}\beta_j w_{ji}|z_{ji}-z_{li}|+\frac{1}{2}(x_i-z_{li})^2. \end{aligned} \tag{40}$$

We can then derive from the expression in (37) the following

$$-\frac{\lambda}{L}\sum_{j=0}^{J}\beta_j w_{ji}\le x_i-z_{li}\le\frac{\lambda}{L}\sum_{j=0}^{J}\beta_j w_{ji}\Leftrightarrow|x_i-z_{li}|\le\frac{\lambda}{L}\sum_{j=0}^{J}\beta_j w_{ji}. \tag{41}$$

Eventually, we observe that the part including v_i in the right hand side of the inequality $h(v_i)$ in (40) is

$$|v_i-z_{li}|\Big(\frac{\lambda}{L}\sum_{j=0}^{J}\beta_j w_{ji}-|x_i-z_{li}|\Big)+\frac{1}{2}(v_i-z_{li})^2. \tag{42}$$

With (41), the expression in (42) is minimum when $v_i = z_{li}$. Therefore, we deduce that $h(v_i)$ in (32) is minimum when $v_i = z_{li}$. ■

In summary, from (35) with the conditions in (36), (37) and Lemma A.2, we obtain (30) as the proof. ■

APPENDIX B
BOUND ON COMPRESSIVE MEASUREMENTS

Measurement Condition. We use notations $\boldsymbol{x}$, $\boldsymbol{v}$, and $\boldsymbol{y}$ without index t in the following proof. We restate the measurement condition in convex optimization [44], which is used in the derivation of the measurement bound. The problem in this paper is an instance of (5), where $f(\boldsymbol{x}):=1/2\|\boldsymbol{\Phi}\boldsymbol{x}-\boldsymbol{y}\|_2^2$ is a smooth convex function and $g(\boldsymbol{x}):=\mathbb{R}^n\to\mathbb{R}$ is a continuous convex function possibly non-smooth. Let us denote that the subdifferential $\partial g(\cdot)$ [55] of a convex function $g(\cdot)$ at a point $\boldsymbol{x} \in \mathbb{R}^n$ is given by $\partial g(\boldsymbol{x}) := \{\boldsymbol{u} \in \mathbb{R}^n : g(\boldsymbol{y}) \ge g(\boldsymbol{x}) + \boldsymbol{u}^{\mathrm{T}}(\boldsymbol{y}-\boldsymbol{x})$ for all $\boldsymbol{y} \in \mathbb{R}^n\}$. Let $\boldsymbol{g} \sim \mathcal{N}(0, \mathbf{I}_n)$ denote a vector of n independent, zero-mean, and unit-variance Gaussian random variables and $\mathbb{E}_{\boldsymbol{g}}[\cdot]$ be the expectation with respect to $\boldsymbol{g}$. The Euclidean distance of $\boldsymbol{g}$ with respect to a convex cone C [44] is defined by

$$\text{dist}(\boldsymbol{g}, C) := \min_{\boldsymbol{u}}\{\|\boldsymbol{g}-\boldsymbol{u}\|_2 : \boldsymbol{u} \in C\}. \tag{43}$$

Proposition B.1. *[44, Corollary 3.3, Proposition 3.6] Let $\boldsymbol{\Phi} \in \mathbb{R}^{m\times n}$ be a random projection, whose elements are randomly drawn from an i.i.d. Gaussian distribution, and λ_m, $\omega(\boldsymbol{x})$ be defined as in Theorem III.1 and [44, Definition 3.1].*

1) *By observing $\boldsymbol{y} = \boldsymbol{\Phi}\boldsymbol{x} \in \mathbb{R}^m$ and solving* (5)*, $\boldsymbol{x} \in \mathbb{R}^n$ is successfully recovered with probability greater than $1 - \exp(-\frac{1}{2}[\lambda_m - \omega(\boldsymbol{x})]^2)$ provided that $m \ge U_g + 1$.*

2) If we observe $\boldsymbol{y} = \boldsymbol{\Phi x} + \boldsymbol{\vartheta}$ with the noise $\boldsymbol{\vartheta}$ that is bounded as $\|\boldsymbol{\vartheta}\|_2 < \sigma$. Let $\hat{\boldsymbol{x}}$ denote any solution in (5) *and $0 < \rho < 1$. We have that $\|\boldsymbol{x} - \hat{\boldsymbol{x}}\|_2 < \frac{2\sigma}{1-\sqrt{\rho}}$ with probability greater than $1 - \exp(-\frac{1}{2}[\lambda_m - \omega(\boldsymbol{x}) - (1-\sqrt{\rho})\sqrt{m}]^2)$ provided that $m \geq \frac{U_g + 3/2}{\rho}$.*

The quantity U_g is calculated given a convex norm function $g(\boldsymbol{x}) := \mathbb{R}^n \to \mathbb{R}$ by

$$U_g = \min_{\tau \geq 0} \mathbb{E}_{\boldsymbol{g}}\Big[\mathrm{dist}^2(\boldsymbol{g}, \tau \cdot \partial g(\boldsymbol{x}))\Big]. \tag{44}$$

Supporting results. Recall that the probability density of the normal distribution $\mathcal{N}(0,1)$ with zero-mean and unit variance $\psi(x)$ is given by

$$\psi(x) := (1/\sqrt{2\pi}) e^{-x^2/2}. \tag{45}$$

We also consider the following inequality [42]:

$$\frac{(1 - x^{-1})}{\sqrt{\pi \log(x)}} \leq \frac{1}{\sqrt{2\pi}} \leq \frac{2}{5}, \tag{46}$$

for all $x > 1$. Moreover, adhering to the formulation in [42], we use the following inequality with $x > 0$ in our derivations:

$$\frac{1}{\sqrt{2\pi}} \int_x^\infty (v - x)^2 e^{-v^2/2} \mathrm{d}v \leq \frac{\psi(x)}{x}. \tag{47}$$

***Proof of Theorem III.1**:* We firstly derive the bound in (28); the bound in (29) is then derived as its noisy-case counterpart. We derive the bounds based on Proposition B.1.

We first compute the subdifferential $\partial g(\boldsymbol{x})$ and then the distance, $\mathrm{dist}(\cdot)$, between the standard normal vector $\boldsymbol{g}$ and $\partial g(\boldsymbol{x})$. The $\boldsymbol{u} \in \partial g(\boldsymbol{x})$ of $g(\boldsymbol{x})$ is derived through the separate components of the sum $g(\boldsymbol{x}) = \sum_{j=0}^J \beta_j g_j(\boldsymbol{x})$, where $g_j(\boldsymbol{x}) = \|\mathbf{W}_j(\boldsymbol{x} - \boldsymbol{z}_j)\|_1$. As a result, $\partial g(\boldsymbol{x}) = \sum_{j=0}^J \beta_j \partial g_j(\boldsymbol{x})$.

Considering the distance from the standard normal vector $\boldsymbol{g}$ to the subdifferential $\partial g(\boldsymbol{x})$ given by (43), we derive a relation between this distance and all separate distances of $g_j(\boldsymbol{x})$ as

$$\mathrm{dist}^2(\boldsymbol{g}, \tau \cdot \partial g(\boldsymbol{x})) \leq \sum_{j=0}^J \beta_j \mathrm{dist}^2(\boldsymbol{g}, \tau \cdot \partial g_j(\boldsymbol{x})), \tag{48}$$

where $\sum_{j=0}^J \beta_j = 1$. Taking the expectation of (48) gives

$$\mathbb{E}_{\boldsymbol{g}}[\mathrm{dist}^2(\boldsymbol{g}, \tau \cdot \partial g(\boldsymbol{x}))] \leq \sum_{j=0}^J \beta_j \mathbb{E}_{\boldsymbol{g}}[\mathrm{dist}^2(\boldsymbol{g}, \tau \cdot \partial g_j(\boldsymbol{x}))]. \tag{49}$$

1) Distance Expectation: We now consider each component $\mathbb{E}_{\boldsymbol{g}}[\mathrm{dist}^2(\boldsymbol{g}, \tau \cdot \partial g_j(\boldsymbol{x}))]$, where $g_j(\boldsymbol{x}) = \|\mathbf{W}_j(\boldsymbol{x} - \boldsymbol{z}_j)\|_1$. We firstly consider $\boldsymbol{x} - \boldsymbol{z}_j$. Without loss of generality, we have assumed that the sparse vector $\boldsymbol{x} - \boldsymbol{z}_j$ has s_j supports or $\|\boldsymbol{x} - \boldsymbol{z}_j\|_0 = s_j$ in the form $\boldsymbol{x} - \boldsymbol{z}_j = (x_1 - z_{j1}, ..., x_{s_j} - z_{js_j}, 0, ..., 0)$. The subdifferential $\boldsymbol{u} \in \partial g_j(\boldsymbol{x})$ of $g_j(\boldsymbol{x})$ is given by:

$$\begin{cases} u_i = w_{ji} \mathrm{sign}(x_i - z_{ji}), & i = 1, ..., s_j \\ |u_i| \leq w_{ji}, & i = s_j + 1, ..., n. \end{cases} \tag{50}$$

Let us consider the weights w_{ji} in Line 10 in Algorithm 2 for a specific $\boldsymbol{z}_j$ and denote that $\eta_j = \frac{n}{\sum_{i=1}^n (|x_i - z_{ji}| + \epsilon)^{-1}}$. We express w_{ji} for the source $\boldsymbol{x} - \boldsymbol{z}_j$ as

$$w_{ji} = \begin{cases} \frac{\eta_j}{|x_i - z_{ji}| + \epsilon}, & i = 1, ..., s_j \\ \frac{\eta_j}{\epsilon}, & i = s_j + 1, ..., n. \end{cases} \tag{51}$$

We can then compute the distance from the standard normal vector $\boldsymbol{g}$ to the subdifferential $\partial g_j(\boldsymbol{x})$ based on (43) as

$$\begin{aligned} \mathrm{dist}^2(\boldsymbol{g}, \tau \cdot \partial g_j(\boldsymbol{x})) = &\sum_{i=1}^{s_j} \Big(g_i - \tau w_{ji} \mathrm{sign}(x_i - z_{ji})\Big)^2 \\ &+ \sum_{i=s_j+1}^{n} \Big(\max(|g_i| - \tau w_{ji}, 0)\Big)^2, \end{aligned} \tag{52}$$

where $\max(a, 0)$ returns the maximum value between $a \in \mathbb{R}$ and 0. Further, after taking the expectation of (52), we obtain

$$\begin{aligned} \mathbb{E}_{\boldsymbol{g}}[\mathrm{dist}^2(\boldsymbol{g}, \tau \cdot \partial g_j(\boldsymbol{x}))] = &s_j + \tau^2 \sum_{i=1}^{s_j} w_{ji}^2 \\ &+ \sqrt{\frac{2}{\pi}} \sum_{i=s_j+1}^{n} \int_{\tau w_{ji}}^\infty (v - \tau w_{ji}) \mathrm{e}^{-v^2/2} \mathrm{d}v. \end{aligned} \tag{53}$$

We apply (47) on the third term in (53) to get

$$\mathbb{E}_{\boldsymbol{g}}[\mathrm{dist}^2(\boldsymbol{g}, \tau \cdot \partial g_j(\boldsymbol{x}))] \leq s_j + \tau^2 \sum_{i=1}^{s_j} w_{ji}^2 + 2 \sum_{i=s_j+1}^{n} \frac{\psi(\tau w_{ji})}{\tau w_{ji}}. \tag{54}$$

2) Bound Derivation: Inserting (54) in (49) for all functions $g_j(\boldsymbol{x})$ gives

$$\begin{aligned} \mathbb{E}_{\boldsymbol{g}}[\mathrm{dist}^2(\boldsymbol{g}, \tau \cdot \partial g(\boldsymbol{x}))] \leq &\sum_{j=0}^J \beta_j s_j + \tau^2 \sum_{j=0}^J \beta_j \sum_{i=1}^{s_j} w_{ji}^2 \\ &+ 2 \sum_{j=0}^J \beta_j (n - s_j) \frac{\psi(\tau \eta_j / \epsilon)}{\tau \eta_j / \epsilon}, \end{aligned} \tag{55}$$

where $w_{ji} = \eta_j / \epsilon$ in (51) are used in the third term of (55). Let us denote $\widehat{\eta} = \min_j \{\eta_j\}$. It can be shown that the function $\psi(x)/x$ decreases with $x > 0$. Thus $\frac{\psi(\tau \eta_j / \epsilon)}{\tau \eta_j / \epsilon} \leq \frac{\psi(\tau \widehat{\eta} / \epsilon)}{\tau \widehat{\eta} / \epsilon}$. Then we can write (55) as

$$\begin{aligned} \mathbb{E}_{\boldsymbol{g}}[\mathrm{dist}^2(\boldsymbol{g}, \tau \cdot \partial g(\boldsymbol{x}))] \leq &\sum_{j=0}^J \beta_j s_j + \tau^2 \sum_{j=0}^J \beta_j \sum_{i=1}^{s_j} w_{ji}^2 \\ &+ 2\Big(n - \sum_{j=0}^J \beta_j s_j\Big) \frac{\psi(\tau \widehat{\eta} / \epsilon)}{\tau \widehat{\eta} / \epsilon}. \end{aligned} \tag{56}$$

Inequality (56) holds due to $\sum_{j=0}^J \beta_j (n - s_j) = n - \sum_{j=0}^J \beta_j s_j$.

We denote $\bar{s} = \sum_{j=0}^J \beta_j s_j$ and from (44), (56), we have

$$U_g \leq \min_{\tau \geq 0} \Big\{ \bar{s} + \tau^2 \sum_{j=0}^J \beta_j \sum_{i=1}^{s_j} w_{ji}^2 + 2(n - \bar{s}) \frac{\psi(\tau \widehat{\eta} / \epsilon)}{\tau \widehat{\eta} / \epsilon} \Big\}. \tag{57}$$

Inserting (45) in (57) gives

$$U_g \leq \min_{\tau \geq 0} \Big\{ \bar{s} + \tau^2 \sum_{j=0}^J \beta_j \sum_{i=1}^{s_j} w_{ji}^2 + (n - \bar{s}) \frac{2}{\sqrt{2\pi}} \frac{e^{(-\tau^2 \widehat{\eta}^2 / 2\epsilon^2)}}{\tau \widehat{\eta} / \epsilon} \Big\}. \tag{58}$$

We can select a parameter $\tau > 0$ in (58) to obtain a bound, here setting $\tau = (\epsilon / \widehat{\eta}) \sqrt{2 \log(n / \bar{s})}$ gives

$$U_g \leq \bar{s} + 2 \log(\frac{n}{\bar{s}}) \frac{\epsilon^2}{\widehat{\eta}^2} \sum_{j=0}^J \beta_j \sum_{i=1}^{s_j} w_{ji}^2 + \frac{\bar{s}(1 - \bar{s}/n)}{\sqrt{\pi \log(n / \bar{s})}}. \tag{59}$$

We denote $\alpha = \frac{\epsilon^2}{\widehat{\eta}^2} \sum_{j=0}^J \beta_j \sum_{i=1}^{s_j} w_{ji}^2$ in the second term in (59). Finally, applying (46) to the last term of (59) gives

$$U_g \leq 2\alpha \log(\frac{n}{\bar{s}}) + \frac{7}{5} \bar{s}. \tag{60}$$

As a result, from Proposition B.1, we get the bound in (28) as

$$m_t \geq 2\alpha \log(\frac{n}{\bar{s}}) + \frac{7}{5}\bar{s} + 1. \quad (61)$$

For slowly-changing $\boldsymbol{v}_t$, we assume that the measurement error $\|\boldsymbol{\Phi}(\hat{\boldsymbol{v}}_t - \boldsymbol{v}_t)\|_2 < \sigma$. Applying Proposition B.1 for the noisy case, we get the bound in (29). ■

REFERENCES

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

[2] H. V. Luong, N. Deligiannis, J. Seiler, S. Forchhammer, and A. Kaup, "Compressive online robust principle component analysis with multiple prior information," in *Proc. of IEEE Global Conf. Signal Inf. Process.*, 2017.

[3] S. Wold, K. Esbensen, and P. Geladi, "Principal component analysis," *Chemometrics and intelligent laboratory systems*, vol. 2, no. 1-3, pp. 37–52, 1987.

[4] 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 *Proc. of Advances in Neural Information Processing Systems*, 2009.

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

[6] V. Chandrasekaran, S. Sanghavi, P. A. Parrilo, and A. S. Willsky, "Rank-sparsity incoherence for matrix decomposition," *SIAM Journal on Optimization*, vol. 21, no. 2, pp. 572–596, 2011.

[7] X. Ding, L. He, and L. Carin, "Bayesian robust principal component analysis," *IEEE Trans. Image Process.*, vol. 20, no. 12, pp. 3419–3430, 2011.

[8] K. Slavakis, G. B. Giannakis, and G. Mateos, "Modeling and optimization for big data analytics: (Statistical) learning tools for our era of data deluge," *IEEE Signal Process. Mag.*, vol. 31, no. 5, pp. 18–31, 2014.

[9] Y. Li, "On incremental and robust subspace learning," *Pattern Recognition*, vol. 37, no. 7, pp. 1509–1518, 2004.

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

[11] G. Mateos and G. B. Giannakis, "Robust PCA as bilinear decomposition with outlier-sparsity regularization," *IEEE Transactions on Signal Processing*, vol. 60, no. 10, pp. 5176–5190, 2012.

[12] 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.

[13] D. Donoho, "Compressed sensing," *IEEE Trans. Inf. Theory*, vol. 52, no. 4, pp. 1289–1306, Apr. 2006.

[14] C. Qiu and N. Vaswani, "Real-time robust principal components' pursuit," in *Proc. of Allerton Conf. Commun., Control, Comput.*, 2010, pp. 591–598.

[15] 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.

[16] P. Rodriguez and B. Wohlberg, "Fast principal component pursuit via alternating minimization," in *Proc. of IEEE Int. Conf. Image Process.*, 2013.

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

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

[19] P. Pan, J. Feng, L. Chen, and Y. Yang, "Online compressed robust PCA," in *Proc. of Int. Joint Conf. on Neural Networks (IJCNN)*. IEEE, 2017, pp. 1041–1048.

[20] A. M. Tekalp, *Digital video processing*. Prentice Hall Press, 2015.

[21] N. Deligiannis, A. Munteanu, T. Clerckx, J. Cornelis, and P. Schelkens, "On the side-information dependency of the temporal correlation in wyner-ziv video coding," in *Proc. of IEEE Int. Conf. on Acoustics, Speech and Signal Process.*, 2009.

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

[23] A. E. Waters, A. C. Sankaranarayanan, and R. Baraniuk, "SpaRCS: Recovering low-rank and sparse matrices from compressive measurements," in *Proc. of Advances in Neural Information Processing Systems*, 2011, pp. 1089–1097.

[24] S. R. Becker, E. J. Candès, and M. C. Grant, "Templates for convex cone problems with applications to sparse signal recovery," *Mathematical Programming Computation*, vol. 3, no. 3, pp. 165–218, 2011.

[25] D. Needell and J. A. Tropp, "CoSaMP: Iterative signal recovery from incomplete and inaccurate samples," *Applied and Computational Harmonic Analysis*, vol. 26, no. 3, pp. 301–321, 2009.

[26] K. Lee and Y. Bresler, "Admira: Atomic decomposition for minimum rank approximation," *IEEE Trans. Inf. Theory*, vol. 56, no. 9, pp. 4402–4416, 2010.

[27] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Foundations and Trends in Machine Learning*, vol. 3, no. 1, pp. 1–122, 2011.

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

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

[30] X. Cao, Q. Zhao, D. Meng, Y. Chen, and Z. Xu, "Robust low-rank matrix factorization under general mixture noise distributions," *IEEE Trans. Image Process.*, vol. 25, no. 10, pp. 4677–4690, Oct 2016.

[31] H. Yong, D. Meng, W. Zuo, and L. Zhang, "Robust online matrix factorization for dynamic background subtraction," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. PP, no. 99, pp. 1–1, Oct 2017.

[32] 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, Sept 2016.

[33] J. F. Mota, N. Deligiannis, A. Sankaranarayanan, V. Cevher, and M. R. Rodrigues, "Dynamic sparse state estimation using ℓ_1-ℓ_1 minimization: Adaptive-rate measurement bounds, algorithms and applications," in *Proc. of IEEE Int. Conf. on Acoustics, Speech and Signal Processing*, Brisbane, Australia, Apr. 2015.

[34] 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.

[35] 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.

[36] 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.

[37] 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.

[38] 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.

[39] V. Cevher, A. Sankaranarayanan, M. F. Duarte, D. Reddy, R. G. Baraniuk, and R. Chellappa, "Compressive sensing for background subtraction," in *Proc. of European Conference on Computer Vision*. Springer, 2008, pp. 155–168.

[40] M. Brand, "Incremental singular value decomposition of uncertain data with missing values," in *Proc. of Europ. Conf. Computer Vision*, 2002.

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

[42] J. F. C. Mota, N. Deligiannis, and M. R. D. Rodrigues, "Compressed sensing with prior information: Strategies, geometry, and bounds," *IEEE Trans. Inf. Theory*, vol. 63, no. 7, pp. 4472–4496, Jul. 2017.

[43] L. Weizman, Y. C. Eldar, and D. B. Bashat, "Compressed sensing for longitunal MRI: An adaptive-weighted approach," *Medical Physics*, vol. 42, no. 9, pp. 5195–5207, 2015.

[44] V. Chandrasekaran, B. Recht, P. A. Parrilo, and A. S. Willsky, "The convex geometry of linear inverse problems," *Foundations of Computational Mathematics*, vol. 12, no. 6, pp. 805–849, 2012.

[45] 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.

[46] 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.
[47] 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.
[48] H. V. Luong, J. Seiler, A. Kaup, and S. Forchhammer, "A reconstruction algorithm with multiple side information for distributed compression of sparse sources," in *Proc. of Data Compr. Conf.*, 2016.
[49] E. Candès, M. B. Wakin, and S. P. Boyd, "Enhancing sparsity by reweighted ℓ_1 minimization," *J. Fourier Anal. Appl.*, vol. 14, no. 5-6, pp. 877–905, 2008.
[50] T. Brox and J. Malik, "Large displacement optical flow: Descriptor matching in variational motion estimation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 3, pp. 500–513, 2011.
[51] The matlab code of the CORPCA algorithm, the test sequences, and the corresponding outcomes. [Online]. Available: https://github.com/huynhlvd/corpca
[52] N. Vaswani, "Stability (over time) of modified-cs for recursive causal sparse reconstruction," in *Proc. of Allerton Conf. Communications, Control, and Computing*, Monticello, Illinois, USA, Sep. 2010.
[53] S. Prativadibhayankaram, H. V. Luong, T. H. Le, and A. Kaup, "Compressive online robust principal component analysis with optical flow for video foreground-background separation," in *Proc. of the Eighth International Symposium on Information and Communication Technology*, Nha Trang City, Viet Nam, Dec. 2017.
[54] M. Dikmen, S. F. Tsai, and T. S. Huang, "Base selection in estimating sparse foreground in video," in *Proc. of IEEE Int. Conf. Image Process.*, 2009.
[55] J.-B. Hiriart-Urruty and C. Lemaréchal, *Fundamentals of Convex Analysis*. Springer, 2004.

Huynh Van Luong (M'09) received the M.Sc. degree in computer engineering with the University of Ulsan, Korea in 2009. He received the Ph.D. degree with the Coding and Visual Communication Group at the Technical University of Denmark, Denmark in 2013.

He was a postdoctoral researcher with Biomedical Research Imaging Center, University of North Carolina at Chapel Hill, USA in 2014. He was awarded the Humboldt Research Fellowship from the Alexander von Humboldt, Germany from 2015 to 2018, where he was with the Chair of Multimedia Communications and Signal Processing (LMS), University of Erlangen-Nuremberg, Germany. He is currently a researcher with the Department of Electronics and Informatics (ETRO), Vrije Universiteit Brussel, Belgium.

His research interests include image and video analysis, compressive sensing, distributed source coding, and high-dimensional data analytics using artificial intelligence.

Nikos Deligiannis (S'08-M'10) is assistant professor with the Electronics and Informatics department at Vrije Universiteit Brussel (VUB) and principal investigator in Data Science at the imec institute in Belgium. He received the Diploma in electrical and computer engineering from University of Patras, Greece, in 2006, and the PhD (Hons.) in applied sciences from Vrije Universiteit Brussel, Belgium, in 2012. From Oct. 2013 to Feb. 2015, he was senior researcher at the Department of Electronic and Electrical Engineering at University College London, UK and technical consultant on big visual data technologies at the British Academy of Film and Television Arts (BAFTA), UK.

His research interests include big data mining, processing and analysis, signal processing, machine learning, internet-of-things networking, and image/video processing and understanding. He has authored over 100 journal and conference publications, book chapters, and two international patent applications.

He has received the 2017 EURASIP Best PhD Award; the 2013 Scientific Prize IBM-FWO Belgium for Informatics; and the Best Paper Award at the 2011 ACM/IEEE International Conference on Distributed Smart Cameras.

Jürgen Seiler (M'09-SM'16) is senior scientist and lecturer at the Chair of Multimedia Communications and Signal Processing at the Friedrich-Alexander Universität Erlangen-Nürnberg, Germany. There, he also received his doctoral degree in 2011 and his diploma degree in Electrical Engineering, Electronics and Information Technology in 2006.

He received the dissertation award of the Information Technology Society of the German Electrical Engineering Association as well as the dissertation award of the Staedtler-Foundation, both in 2012. In 2007, he received diploma awards from the Institute of Electrical Engineering, Electronics and Information Technology, Erlangen, as well as from the German Electrical Engineering Association. He also received scholarships from the German National Academic Foundation and the Lucent Technologies Foundation. He is the co-recipient of four best paper awards and he has authored or co-authored more than 70 technical publications.

From 2012 to 2016, he was member of the Management Committee of EU COST Action IC1105 3D-ConTourNet - "3D Content Creation, Coding and Transmission over Future Media Networks". For this Action, he also served as coordinator for Short Term Scientific Missions. In 2016, he received an LFUI-Guest Professorship at the University of Innsbruck.

His research interests include image and video signal processing, signal reconstruction and coding, signal transforms, and linear systems theory.

Søren Forchhammer (M'04) received the M.S. degree in engineering and the Ph.D. degree from the Technical University of Denmark, Lyngby, in 1984 and 1988, respectively.

Currently, he is a Professor with DTU Fotonik, Technical University of Denmark. He is Head of the Coding and Visual Communication Group. His main interests include source coding, image and video coding, distributed source coding, distributed video coding, compressive sensing, processing for image displays, two-dimensional information theory, visual communications and coding for optical communication.

André Kaup (M'96-SM'99-F'13) received the Dipl.-Ing. and Dr.-Ing. degrees in electrical engineering from Rheinisch-Westfälische Technische Hochschule (RWTH) Aachen University, Aachen, Germany, in 1989 and 1995, respectively.

He was with the Institute for Communication Engineering, RWTH Aachen University, from 1989 to 1995. He joined the Networks and Multimedia Communications Department, Siemens Corporate Technology, Munich, Germany, in 1995 and became Head of the Mobile Applications and Services Group in 1999. Since 2001 he has been a Full Professor and the Head of the Chair of Multimedia Communications and Signal Processing, University of Erlangen- Nuremberg, Erlangen, Germany. From 1997 to 2001 he was the Head of the German MPEG delegation. From 2005 to 2007 he was a Vice Speaker of the DFG Collaborative Research Center 603. From 2015 to 2017 he served as Head of the Department of Electrical Engineering and Vice Dean of the Faculty of Engineering. He has authored around 350 journal and conference papers and has over 70 patents granted or pending. His research interests include image and video signal processing and coding, and multimedia communication.

André Kaup is a member of the IEEE Multimedia Signal Processing Technical Committee, a member of the scientific advisory board of the German VDE/ITG, and a Fellow of the IEEE. He served as an Associate Editor for IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY and was a Guest Editor for IEEE JOURNAL OF SELECTED TOPICS IN SIGNAL PROCESSING. From 1998 to 2001 he served as an Adjunct Professor with the Technical University of Munich, Munich. He was a Siemens Inventor of the Year 1998 and received the 1999 ITG Award. He has received several best paper awards, including the Paul Dan Cristea Special Award from the International Conference on Systems, Signals, and Image Processing in 2013. His group won the Grand Video Compression Challenge at the Picture Coding Symposium 2013 and he received the teaching award of the Faculty of Engineering in 2015.