

Article

Compressive Online Video Background-Foreground Separation Using Multiple Prior Information and Optical Flow

Srivatsa Prativadibhayankaram^{1,†,✉}, Huynh Van Luong^{1,*,‡,✉}, Thanh Ha Le², and André Kaup¹

¹ Multimedia Communications and Signal Processing, University of Erlangen-Nuremberg, 91058 Erlangen, Germany; srivatsa.pv@live.com; huynh.luong@fau.de; andre.kaup@fau.de

² Human Machine Interaction, University of Engineering and Technology, Vietnam National University, Hanoi, Vietnam; ltha@vnu.edu.vn

* Correspondence: H. V. Luong, Email: hvanluon@etrovub.be, Tel: + 32 2 629 1674, Fax: + 32 2 629 2883.

† Current address: Digital Cinema Group, Fraunhofer IIS, Erlangen, Germany

‡ Current address: Department of Electronics and Informatics, Vrije Universiteit Brussel, Brussels, Belgium; hvanluon@etrovub.be



Abstract: In the context of video background-foreground separation, we propose a compressive online Robust Principal Component Analysis (RPCA) with optical flow that separates recursively a sequence of video frames into foreground (sparse) and background (low-rank) components. This separation method can process per video frame from a small set of measurements, in contrast to conventional batch-based RPCA, which processes the full data. The proposed method also leverages multiple prior information by incorporating previously separated background and foreground frames in an $n\text{-}\ell_1$ minimization problem. Moreover, optical flow is utilized to estimate motions between the previous foreground frames and then compensate the motions to achieve higher quality prior foregrounds for improving the separation. Our method is tested on several video sequences in different scenarios for online background-foreground separation given compressive measurements. The visual and quantitative results show that the proposed method outperforms other existing methods.

Keywords: robust principal component analysis; video separation; compressive measurements; prior information; optical flow; motion estimation; motion compensation

1. Introduction

Emerging applications in surveillance and autonomous driving are challenging the existing visual systems to detect and understand objects from visual observations. Video background-foreground separation is one of great important components for object detection, identification, and tracking. In video separation, a video sequence can be separated into a slowly-changing background (modeled by L as a low-rank component) and the foreground (modeled by S , which is a sparse component). RPCA [1,2] was shown to be a robust method for separating the low-rank and sparse components. RPCA decomposes a data matrix M into the sum of unknown sparse S and low-rank L by solving the Principal Component Pursuit (PCP) [1] problem:

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

where $\|\cdot\|_*$ is the matrix nuclear norm (sum of singular values) and $\|\cdot\|_1$ is the ℓ_1 -norm. Many applications of RPCA can be found in computer vision, web data analysis, and recommender systems. However, batch RPCA processes all data samples, e.g., all frames in a video, which demands high computational and memory requirements.

Moreover, the video separation can be improved by taking into account the correlation between consecutive frames. The correlations can be obtained in the form of motions, which manifest as change in information from one frame to the other. Detecting motion is an integral part of the human visual system. A popular and convenient method for estimating motion in computer vision is by using optical flow [3–5] by variational methods. The optical flow estimates motion vectors of all pixels in a given frame due to the relative motions between frames. In particular, the motion vectors at each pixel can be estimated by minimizing a gradient-based matching of pixel gray value, combined with a smoothness criteria [3]. Thereafter, the computed motion vectors in the horizontal and vertical directions [4] are used to compensate and predict information in the next frame. For producing highly accurate motions and accurate correspondences between frames, a large displacement optical flow [6], that combines a coarse-to-fine optimization with descriptor matching can be used to estimate the motions from previously separated frames and subsequently use them to support the separation of the current frame.

In order to deal with the video separation in an online manner, we consider an online RPCA algorithm that recursively processes a sequence of frames (a.k.a., the column-vectors in M) per time instance. Additionally, we aim at recovering the foreground and background from a small set of measurements rather than a full frame data, leveraging information from a set of previously separated frames. In particular, at time instance t , we wish to separate M_t into $S_t = [x_1 \ x_2 \ \dots \ x_t]$ and $L_t = [v_1 \ v_2 \ \dots \ v_t]$, where $[\cdot]$ denotes a matrix and $x_t, v_t \in \mathbb{R}^n$ are column-vectors in S_t and L_t , respectively. We assume that $S_{t-1} = [x_1 \ x_2 \ \dots \ x_{t-1}]$ and $L_{t-1} = [v_1 \ v_2 \ \dots \ v_{t-1}]$ have been recovered at time instance $t-1$. In the next instance t , we have access to compressive measurements of the full frame, a.k.a., vector $x_t + v_t$, that is, we observe $y_t = \Phi(x_t + v_t)$, where $\Phi \in \mathbb{R}^{m \times n}$ ($m < n$) is a random projection. The recovery problem at time instance t is thus written [8] as

$$\min_{x_t, v_t} \| [L_{t-1} \ v_t] \|_* + \lambda \| x_t \|_1 \text{ subject to } y_t = \Phi(x_t + v_t), \quad (2)$$

where L_{t-1} and S_{t-1} are given.

1.1. Related Work

Many methods [9–15] have been proposed to solve the separation problem by advancing RPCA [1] exist. Incremental PCP [9] processes each column-vector in M at a time, but it needs access to complete data (e.g., full frames) rather than compressive data. A counterpart of batch RPCA that operates on compressive measurements known as Compressive PCP can be found in [16]. The studies in [12–15,17] aim at solving the problem of online estimation of low-dimensional subspaces from randomly subsampled data for modeling the background. An algorithm to recover the sparse component x_t in (2) has been proposed in [18], however, the low-rank component v_t in (2) is not recovered per time instance from a small number of measurements.

The alternative method in [19,20] estimates the number of compressive measurements required to recover foreground x_t per time instance via assuming the background v_t not-varying. This assumption is invalid in realistic scenarios due to variations in illumination or dynamic backgrounds. The online method in [17] works on compressive measurements without taking the prior information into account.

The problem of separating a sequence of time-varying frames using prior information brings significant improvements in the context of online RPCA [18,21,22]. Some studies on recursive recovery from low-dimensional measurements have been proposed to leverage prior information [18,19,21,23]. The study in [23] provided a comprehensive overview of the domain, reviewing a class of recursive algorithms. A review of all the recent problem formulations is made in [24] with a unified view with Decomposition into Low-rank plus Additive Matrices (DLAM).

The studies in [18,21] used modified-CS [25] to leverage prior knowledge under the condition of slowly varying support and signal values. However, this method as well as the methods in [12,13,15] do not explore the correlations between the current frame and multiple previously separated frames.

The recent work in [8] leverages correlations across the previously separated foreground frames. However, displacements between the previous foreground frames and the current frame are not taken into account. These displacements can incur the degradation of the separation performance.

There are alternative approaches such as, double-constrained RPCA, namely shape and confidence map based RPCA (SCM-RPCA) in [26] that combines saliency map along with RPCA for automated maritime surveillance. Recently a method that incorporates the spatial and temporal sparse subspace clustering into the RPCA framework was developed in [27] and earlier in [28]. Recent studies in [29] proposes a non-parametric approach for background subtraction.

1.2. Contributions

In this paper, we propose a *compressive online robust PCA with optical flow* (CORPCA-OF¹) method, which is based on our previous work in [8]. We leverage information from previously separated foreground frames via optical flow [6]. The novelty of CORPCA-OF over CORPCA [8] is that we make use of optical flow to estimate and compensate motions between the foreground frames, in order to generate new prior foreground frames. These new prior frames have high correlation with the current frame and thus improve the separation. We also exploit the slowly-changing characteristics of backgrounds known as low-rank components via an incremental SVD [30] method. The compressive separation problem in (2) is solved in online manner by minimizing not only an n - ℓ_1 -norm cost function [31] for the sparse foreground but also the rank of a matrix for the low-rank backgrounds. Thereafter, the new separated foreground and background frames are used to update the prior knowledge for the next processing instance. This method makes video separation better with higher accuracy and more efficient implementation. The algorithm has been implemented in C++ using OpenCV and also tested on Matlab. Various functions/methods that are needed in performing video separation or general compressed sensing have been developed in C++ and can be used as stand-alone libraries.

The paper is organized as follows. We provide a brief introduction to the CORPCA algorithm [8], on which our proposed CORPCA-OF is built upon. Based on this, we formulate the problem statement in Sec. 2.1, followed by a proposed CORPCA-OF algorithm in Sec. 2.2.1. The visual and quantitative results obtained by testing our method on real video sequences are presented and discussed in Sec. 3.

2. Compressive Online Robust PCA Using Multiple Prior Information and Optical Flow

In this section, we firstly review the CORPCA [8] algorithm for online compressive video separation and state our problem. Thereafter, we proposed CORPCA-OF, which is summarized in the proposed CORPCA-OF algorithm.

2.1. Compressive Online Robust PCA (CORPCA) for video separation

The CORPCA algorithm [8] for video separation is based on RAMSIA [31], that solves an n - ℓ_1 minimization problem with adaptive weights to recover a sparse signal \mathbf{x} from low-dimensional random measurements $\mathbf{y} = \Phi\mathbf{x}$ with the aid of multiple prior information or side information \mathbf{z}_j , $j \in \{0, 1, \dots, J\}$, with $\mathbf{z}_0 = \mathbf{0}$. The objective function of RAMSIA [31] is given by

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

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

¹ Our initial work has been reported at [7]

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

The objective function of CORPCA is to solve Problem (2) and can be formulated by

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

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

The CORPCA algorithm² solves Problem (4) given that \mathbf{Z}_{t-1} and \mathbf{B}_{t-1} are known (they are obtained from the time instance or recursion). Thereafter, we update \mathbf{Z}_t and \mathbf{B}_t , which are used in the subsequent time instance.

Let us denote $f(\mathbf{v}_t, \mathbf{x}_t) = (1/2) \|\Phi(\mathbf{x}_t + \mathbf{v}_t) - \mathbf{y}_t\|_2^2$, $g(\mathbf{x}_t) = \lambda \sum_{j=0}^J \beta_j \|\mathbf{W}_j(\mathbf{x}_t - \mathbf{z}_j)\|_1$, and $h(\mathbf{v}_t) = \left\| [\mathbf{B}_{t-1} \mathbf{v}_t] \right\|_*$, where $f(\cdot)$ is a function of both background and foreground, $g(\cdot)$ is a function of the foreground (sparse) component and $h(\cdot)$ is a function of the background (low rank) component. The components $\mathbf{x}_t^{(k+1)}$ and $\mathbf{v}_t^{(k+1)}$ are solved iteratively at iteration $k+1$ via the soft thresholding operator [32] for \mathbf{x}_t and the single value thresholding operator [33] for \mathbf{v}_t :

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

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

2.2. Video Foreground and Background Separation using CORPCA-OF

Problem statement. Using the prior information in CORPCA [8] has provided significant improvement of the current frame separation. However, there can be displacements between the consecutive frames that can deteriorate the separation performance. Fig. 1 illustrates an example of three previous foreground frames, \mathbf{x}_{t-3} , \mathbf{x}_{t-2} , and \mathbf{x}_{t-1} . These frames can be used directly as prior information to recover foreground \mathbf{x}_t and background \mathbf{v}_t due to temporal correlations between \mathbf{x}_t and \mathbf{x}_{t-3} , \mathbf{x}_{t-2} , \mathbf{x}_{t-1} , as seen in CORPCA. In the last row of the prior foreground frames in Fig. 1, it can be seen that motions exist between frames. By estimating motion using optical flow [6], we can obtain motions between the previous foreground frames as in Fig. 1, which are visualized using color codes based on the magnitude and direction of motion vectors [6]. These motions can be compensated to generate better quality prior frames (compare compensated frames \mathbf{x}'_{t-3} , \mathbf{x}'_{t-2} with \mathbf{x}_{t-3} , \mathbf{x}_{t-2} , \mathbf{x}_{t-1}), and it is better correlated to \mathbf{x}_t . In this work, we discuss an algorithm we proposed - CORPCA with Optical Flow (CORPCA-OF), whose work flow is shown in Fig. 1. Optical flow [6] is used to improve prior foreground frames.

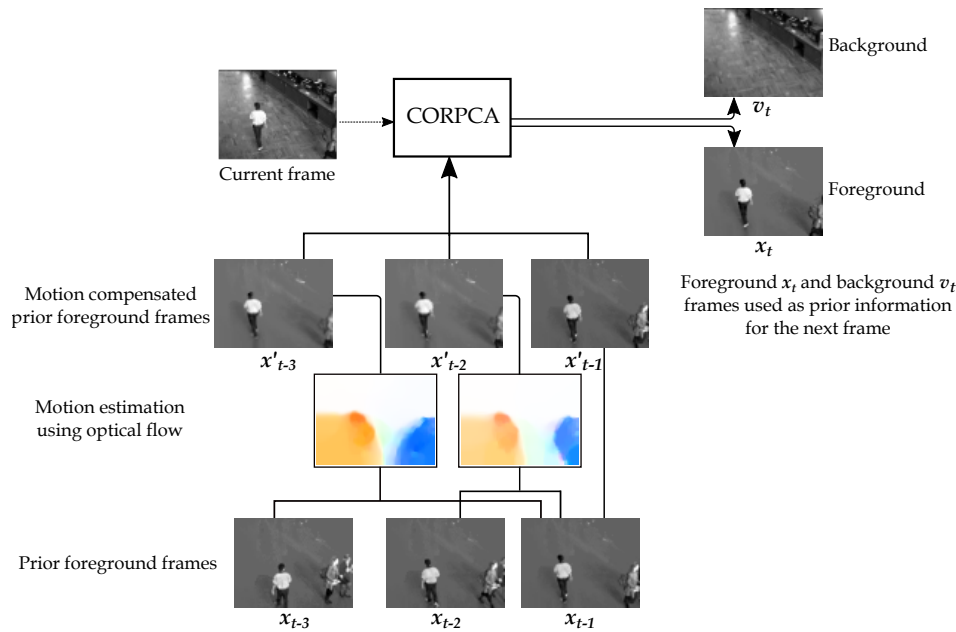


Figure 1. CORPCA-OF work flow.

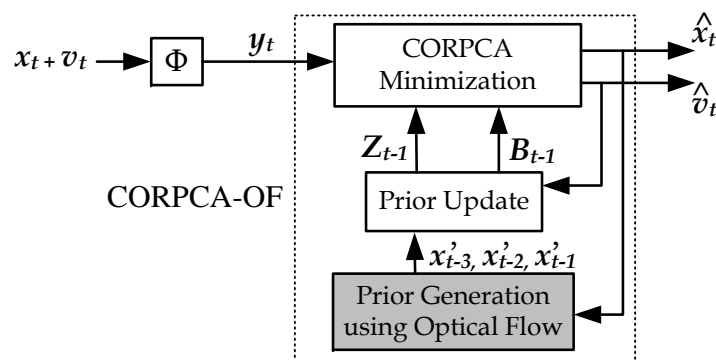


Figure 2. Compressive separation model using CORPCA-OF.

2.2.1. Compressive Separation Model with CORPCA-OF

A compressive separation model using the CORPCA-OF method is shown in Fig. 2. At a time instance t , the inputs consist of compressive measurements $y_t = \Phi(x_t + v_t)$ and prior information from time instance $t - 1$, Z_{t-1} (foreground) and B_{t-1} (background). The model outputs foreground and background information x_t and v_t by solving the CORPCA minimization problem in (4). Finally, the outputs x_t and v_t are used to generate better prior foreground information via a prior generation using optical flow and update Z_{t-1} and B_{t-1} for the next instance via a prior update. The novel block of CORPCA-OF compared with CORPCA [8] is the Prior Generation using Optical Flow, where prior foreground information is improved by exploiting the correlation between frames using large displacement optical flow [6]. The method is further described in Algorithm 1.

² The code of the CORPCA algorithm, the test sequences, and the corresponding outcomes are available at <https://github.com/huynhlvd/corpc>

Algorithm 1: The proposed CORPCA-OF algorithm.

Input: y_t, Z_{t-1}, B_{t-1} ;
Output: $\hat{x}_t, \hat{v}_t, Z_t, B_t$;
 // Initialize variables and parameters.
 1 $x_t^{(-1)} = x_t^{(0)} = \mathbf{0}; v_t^{(-1)} = v_t^{(0)} = \mathbf{0}; \xi_{-1} = \xi_0 = 1; \mu_0 = 0; \bar{\mu} > 0; \lambda > 0; 0 < \epsilon < 1; k = 0; g_1(\cdot) = \|\cdot\|_1$;
 // Motion estimation and compensation using Large Displacement Optical Flow [6]
 2 $(v_{1x}, v_{1y}) = f_{ME}(x_{t-1}, x_{t-2})$;
 3 $(v_{2x}, v_{2y}) = f_{ME}(x_{t-1}, x_{t-3})$;
 4 $x'_{t-1} = x_{t-1}$;
 5 $x'_{t-2} = f_{MC}(x_{t-1}, v_{1x}, v_{1y})$;
 6 $x'_{t-3} = f_{MC}(x_{t-1}, \frac{1}{2}v_{2x}, \frac{1}{2}v_{2y})$;
 7 $z_J = x'_{t-1}; z_{J-1} = x'_{t-2}; z_{J-2} = x'_{t-3}$;
 8 **while not converged do**
 // Solve Problem (4).
 9 $\tilde{v}_t^{(k)} = v_t^{(k)} + \frac{\xi_{k-1}-1}{\xi_k}(v_t^{(k)} - v_t^{(k-1)})$;
 10 $\tilde{x}_t^{(k)} = x_t^{(k)} + \frac{\xi_{k-1}-1}{\xi_k}(x_t^{(k)} - x_t^{(k-1)})$;
 11 $\nabla_{v_t} f(\tilde{v}_t^{(k)}, \tilde{x}_t^{(k)}) = \nabla_{x_t} f(\tilde{v}_t^{(k)}, \tilde{x}_t^{(k)}) = \Phi^T(\Phi(\tilde{v}_t^{(k)} + \tilde{x}_t^{(k)}) - y_t)$;
 12 $(U_t, \Sigma_t, V_t) = \text{incSVD}\left(\left[B_{t-1} \left(\tilde{v}_t^{(k)} - \frac{1}{2}\nabla_{v_t} f(\tilde{v}_t^{(k)}, \tilde{x}_t^{(k)})\right)\right]\right)$;
 13 $\Theta_t = U_t \Gamma_{\frac{\mu_k}{2} g_1}(\Sigma_t) V_t^T$;
 14 $v_t^{(k+1)} = \Theta_t(:, \text{end})$;
 15 $x_t^{(k+1)} = \Gamma_{\frac{\mu_k}{2} g}(\tilde{x}_t^{(k)} - \frac{1}{2}\nabla_{x_t} f(\tilde{v}_t^{(k)}, \tilde{x}_t^{(k)}))$; where $\Gamma_{\frac{\mu_k}{2} g}(\cdot)$ is given as in RAMSIA [31];
 // Compute the updated weights [31].
 16 $w_{ji} = \frac{n(|x_{ti}^{(k+1)} - z_{ji}| + \epsilon)^{-1}}{\sum_{l=1}^n (|x_{tl}^{(k+1)} - z_{jl}| + \epsilon)^{-1}}$;
 17 $\beta_j = \frac{\left(\|W_j(x_t^{(k+1)} - z_j)\|_1 + \epsilon\right)^{-1}}{\sum_{l=0}^J \left(\|W_l(x_t^{(k+1)} - z_l)\|_1 + \epsilon\right)^{-1}}$;
 18 $\xi_{k+1} = (1 + \sqrt{1 + 4\xi_k^2})/2; \mu_{k+1} = \max(\epsilon\mu_k, \bar{\mu})$;
 19 $k = k + 1$;
 20 **end**
 // Update prior information.
 21 $Z_t := \{z_j = x_{t-J+j}^{(k+1)}\}_{j=1}^J$;
 22 $B_t = U_t(:, 1:d) \Gamma_{\frac{\mu_k}{2} g_1}(\Sigma_t) (1:d, 1:d) V_t(:, 1:d)^T$;
 23 **return** $\hat{x}_t = x_t^{(k+1)}, \hat{v}_t = v_t^{(k+1)}, Z_t, B_t$;

2.2.2. Prior Generation using Optical Flow

The main idea of CORPCA-OF is to improve the foreground prior frames using the correlation between frames, which is done by estimating motion between frames via optical flow. In Algorithm 1, the prior frames are initialized with x_{t-1} , x_{t-2} and x_{t-3} . Optical flow is used to compute the motions between frames x_{t-1} and x_{t-2} (also x_{t-1} and x_{t-3}) to obtain flow vectors for these two frames. This can be seen in Fig. 1 from the color coded representation of optical flow fields [6]. The function $f_{ME}(\cdot)$ in Lines 2 and 3 [see Algorithm 1] computes the motions between prior foreground frames. It involves computing the optical flow vectors consisting of horizontal (x) and vertical (y) components, and is denoted by v_{1x}, v_{2x} and $v_{1y}, v_{2y} \in \mathbb{R}^n$, respectively. The estimated motions in the form of optical flow vectors, (v_{1x}, v_{1y}) and (v_{2x}, v_{2y}) , are then used to predict the following frames by compensating for

the forward motions on x_{t-1} . The prior frames, x'_{t-2} and x'_{t-3} are generated by performing motion compensation. This is indicated by the function $f_{MC}(\cdot)$ as shown by Lines 5 and 6 in Algorithm 1.

Considering a point or a pixel i in the given frame, the horizontal and vertical components v_{1xi} and v_{1yi} of corresponding horizontal and vertical flow vectors \mathbf{v}_{1x} and \mathbf{v}_{1y} are obtained, as mentioned in [34], by solving :

$$I_{1x} \cdot v_{1xi} + I_{1y} \cdot v_{1yi} + I_{1t} = 0, \quad (7)$$

where $I_{1x} = \partial I_1 / \partial x$ and $I_{1y} = \partial I_1 / \partial y$ are the intensity changes in the horizontal (x) and vertical (y) directions, respectively, constituting the spatial gradients of the intensity level I_1 ; $I_{1t} = \partial I_1 / \partial t$ is the time gradient, which is a measure of temporal change in the intensity level at point i . There are various methods [3–6] to determine v_{1xi} and v_{1yi} . Our solution is based on large displacement optical flow [6], that is a combination of global and local approaches to estimate all kinds of motion. It involves optimization and minimization of error by using descriptor matching and continuation method, which utilizes feature matching along with conventional optical flow estimation to obtain the flow field. We combine the optical flow components of each point i in the image into two vectors $(\mathbf{v}_{1x}, \mathbf{v}_{1y})$, i.e., the horizontal and the vertical components of the optical flow vector. Similarly, we obtain $(\mathbf{v}_{2x}, \mathbf{v}_{2y})$.

The estimated motions in the form of optical flow vectors are used along with the frame x_{t-1} to produce new prior frames that form the updated prior information. Linear interpolation is used to generate new frames via column interpolation and row interpolation. This is represented as $f_{MC}(\cdot)$ in Lines 5 and 6 in the Algorithm 1. The obtained frame is the result of using the flow fields $(\mathbf{v}_{1x}, \mathbf{v}_{1y})$ and $(\frac{1}{2}\mathbf{v}_{2x}, \frac{1}{2}\mathbf{v}_{2y})$ to predict motions in the next frame and compensate them on the frame x_{t-1} to obtain x'_{t-2} and x'_{t-3} respectively. It should be noted that x'_{t-3} is obtained by compensating for half the motions, i.e., $(\frac{1}{2}\mathbf{v}_{2x}, \frac{1}{2}\mathbf{v}_{2y})$, between x_{t-1} and x_{t-3} . These improved frames x'_{t-2} , x'_{t-3} are more correlated to the current frame x_t than x_{t-2} , x_{t-3} , i.e., without motion estimation and compensation. We also keep the most recent frame $x'_{t-1} = x_{t-1}$ (in Line 4) as one of the prior frames.

Thereafter, $v_t^{(k+1)}$ and $x_t^{(k+1)}$ are iteratively computed as in Lines 14-15 in Algorithm 1. It can be noted that the proximal operator $\Gamma_{\tau g_1}(\cdot)$ in Line 13 of Algorithm 1 is defined [8] as:

$$\Gamma_{\tau g_1}(\mathbf{X}) = \arg \min_{\mathbf{V}} \left\{ \tau g_1(\mathbf{V}) + \frac{1}{2} \|\mathbf{V} - \mathbf{X}\|_2^2 \right\}, \quad (8)$$

where $g_1(\cdot) = \|\cdot\|_1$ (ℓ_1 -norm). The weights \mathbf{W}_j and β_j are updated per iteration of the algorithm (see Lines 16-17). As suggested in [2], the convergence of Algorithm 1 in Line 8 is determined by evaluating the criterion $\|\partial H(x_t, v_t)\|_{x_t^{(k+1)}, v_t^{(k+1)}}^2 < 2 * 10^{-7} \|x_t^{(k+1)}, v_t^{(k+1)}\|_2^2$. In the next step, we perform an update of the priors \mathbf{Z}_t and \mathbf{B}_t .

2.2.3. Prior Update

The update of \mathbf{Z}_t and \mathbf{B}_t [8] is carried out after each time instance (see Lines 21-22, Algorithm 1). Due to the correlation between subsequent frames, we update the prior information \mathbf{Z}_t by using the J latest recovered sparse components, which is given by, $\mathbf{Z}_t := \{z_j = x_{t-J+j}\}_{j=1}^J$. For $\mathbf{B}_t \in \mathbb{R}^{n \times d}$, we consider an adaptive update, which operates on a fixed or constant number d of the columns of \mathbf{B}_t . To this end, the incremental singular value decomposition [30] method (incSVD(\cdot)) in Line 12, Algorithm 1) is used. It is worth noting that the update $\mathbf{B}_t = \mathbf{U}_t \Gamma_{\frac{\mu_k}{2} g_1}(\Sigma_t) \mathbf{V}_t^T$, causes the dimension of \mathbf{B}_t to increase as $\mathbf{B}_t \in \mathbb{R}^{n \times (d+1)}$ after each instance. However, in order to maintain a reasonable number of d , we take $\mathbf{B}_t = \mathbf{U}_t(:, 1:d) \Gamma_{\frac{\mu_k}{2} g_1}(\Sigma_t)(1:d, 1:d) \mathbf{V}_t(:, 1:d)^T$. The computational cost of incSVD(\cdot) is lower than conventional SVD [9,30] since we only compute the full SVD of the middle matrix with size $(d+1) \times (d+1)$, where $d \ll n$, instead of $n \times (d+1)$. The computation of incSVD(\cdot) is presented as follows: The goal is to compute incSVD $[\mathbf{B}_{t-1} \mathbf{v}_t]$, i.e., $[\mathbf{B}_{t-1} \mathbf{v}_t] = \mathbf{U}_t \Sigma_t \mathbf{V}_t^T$. By taking the SVD of $\mathbf{B}_{t-1} \in \mathbb{R}^{n \times d}$ to obtain $\mathbf{B}_{t-1} = \mathbf{U}_{t-1} \Sigma_{t-1} \mathbf{V}_{t-1}^T$. Therefore, we can derive $(\mathbf{U}_t, \Sigma_t, \mathbf{V}_t)$ via $(\mathbf{U}_{t-1}, \Sigma_{t-1}, \mathbf{V}_{t-1})$ and \mathbf{v}_t . We write the matrix $[\mathbf{B}_{t-1} \mathbf{v}_t]$ as

$$[B_{t-1} \ v_t] = \left[U_{t-1} \ \frac{\delta_t}{\|\delta_t\|_2} \right] \cdot \begin{bmatrix} \Sigma_{t-1} & e_t \\ 0^T & \|\delta_t\|_2 \end{bmatrix} \cdot \begin{bmatrix} V_{t-1}^T & 0 \\ 0^T & 1 \end{bmatrix}, \quad (9)$$

where $e_t = U_{t-1}^T v_t$ and $\delta_t = v_t - U_{t-1} e_t$. By computing the SVD of the central term of (9), we obtain $\begin{bmatrix} \Sigma_{t-1} & e_t \\ 0^T & \|\delta_t\|_2 \end{bmatrix} = \tilde{U} \tilde{\Sigma} \tilde{V}^T$. Eventually, we obtain $U_t = \left[U_{t-1} \ \frac{\delta_t}{\|\delta_t\|_2} \right] \cdot \tilde{U}$, $\Sigma_t = \tilde{\Sigma}$, and also, $V_t = \begin{bmatrix} V_{t-1}^T & 0 \\ 0^T & 1 \end{bmatrix} \cdot \tilde{V}$.

3. Experimental Results

In this section we present and discuss the experimental results obtained by applying our method CORPCA-OF on real video data. We also evaluate the performance of our algorithm and compare the results with other existing methods.

Experimental setup. The experiments were carried out on two computers. The Matlab implementation was carried out and tested on a desktop PC with Intel i5 3.5GHz CPU (4 cores) and 12GB RAM. The C++ implementation was carried out and tested on a computer with i7-4510U 2.0 GHz CPU (2 cores) and 8GB memory.

For the experimental evaluation with the various existing methods, mainly two sequences [35], Bootstrap (80 × 60) and Curtain (80 × 64) were used. The Bootstrap sequence consists of 3055 frames and has a static background and a complex foreground. The Curtain sequence contains 2964 frames with a dynamic background and simple foreground motion. For separating each of these sequences, 100 frames are randomly selected and used for initialization of prior information. The prior information is later updated by selecting three most recent frames as seen in Sec. 2.2.3.

We evaluate the performance of the proposed CORPCA-OF in Algorithm 1 and compare it against the existing methods, RPCA [1], GRAFTA [12], and ReProCS [18,23]. RPCA [1] is a batch-based method assuming full access to the data, while GRAFTA [12] and ReProCS [18] are online methods that can recover either the (low-rank) background component (GRASTA) or the (sparse) foreground component (ReProCS) from compressive measurements.

3.1. Prior Information Evaluation

We evaluate the prior information of CORPCA-OF compared with that of CORPCA[8] using the previously separated foreground frames directly. For CORPCA-OF, we generate the prior information by estimating and compensating motions among the previous foreground frames. Fig. 3 shows a few examples of the prior information generated for the sequences Bootstrap and Curtain. In Fig. 3(a), it can be observed that frames #2210', #2211' and #2212' (of CORPCA-OF) are better than corresponding #2210, #2211 and #2212 (of CORPCA) for the current frame #2213, similarly in Figs. 3(b), 3(c), and 3(d). Especially in Fig. 3(c), the generated frames #448' and #449' have significantly improved due to dense motion compensations. In Fig. 3(d), it is clear that the movements of the person is well compensated in #2771' and #2772' by CORPCA-OF compared to #2771 and #2772 respectively, of CORPCA, leading to better correlations with the foreground of current frame #2774. Replace one of the curtain and bootstrap sequences by an elevator and a fountain sequence.

3.2. Compressive Video Foreground and Background Separation

We assess our CORPCA-OF method in the application of compressive video separation and compare it against the existing methods, CORPCA[8], RPCA [1], GRAFTA [12], and ReProCS [18]. We run all methods on the test video sequences. In this experiment, we use $d = 100$ frames as training vectors for the proposed CORPCA-OF, CORPCA[8] as well as for GRAFTA [12] and ReProCS [18]. Three latest previous foregrounds are used as the foreground prior for CORPCA. Meanwhile, CORPCA-OF refines them using optical flow [6].

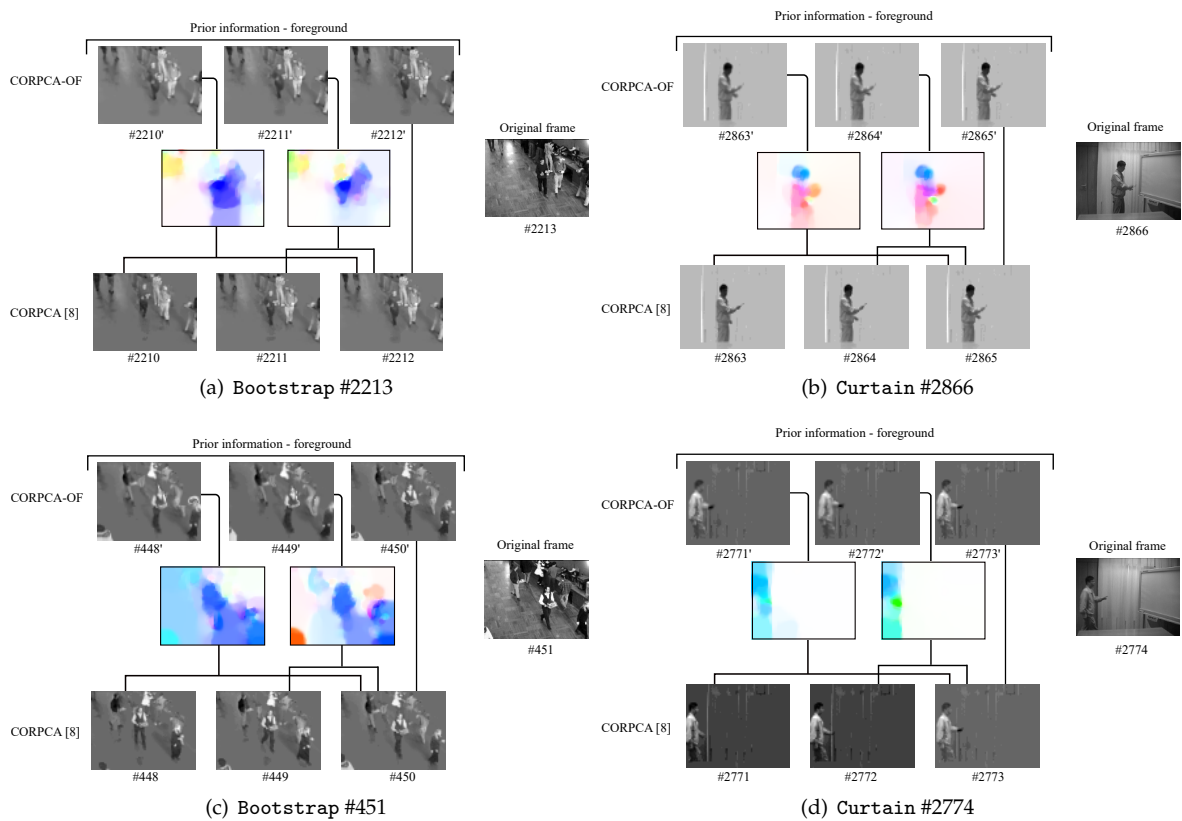


Figure 3. Prior information generation in CORPCA-OF using optical flow [6].

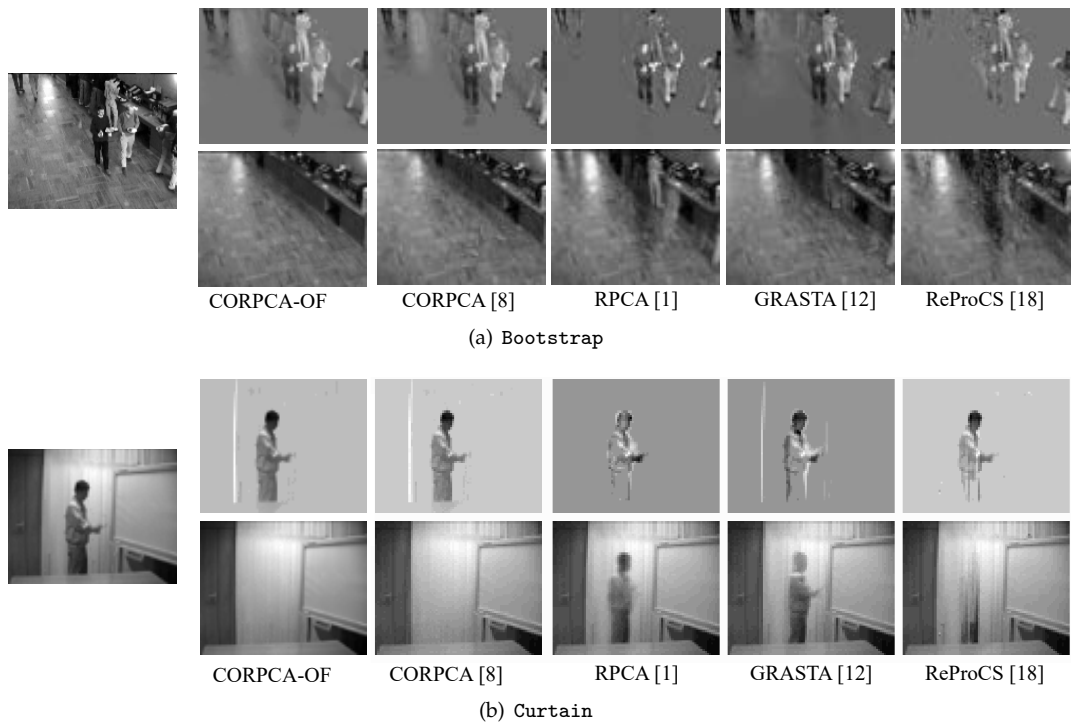


Figure 4. Foreground and background separation for the different separation methods with full data access Bootstrap #2213 and Curtain #2866.

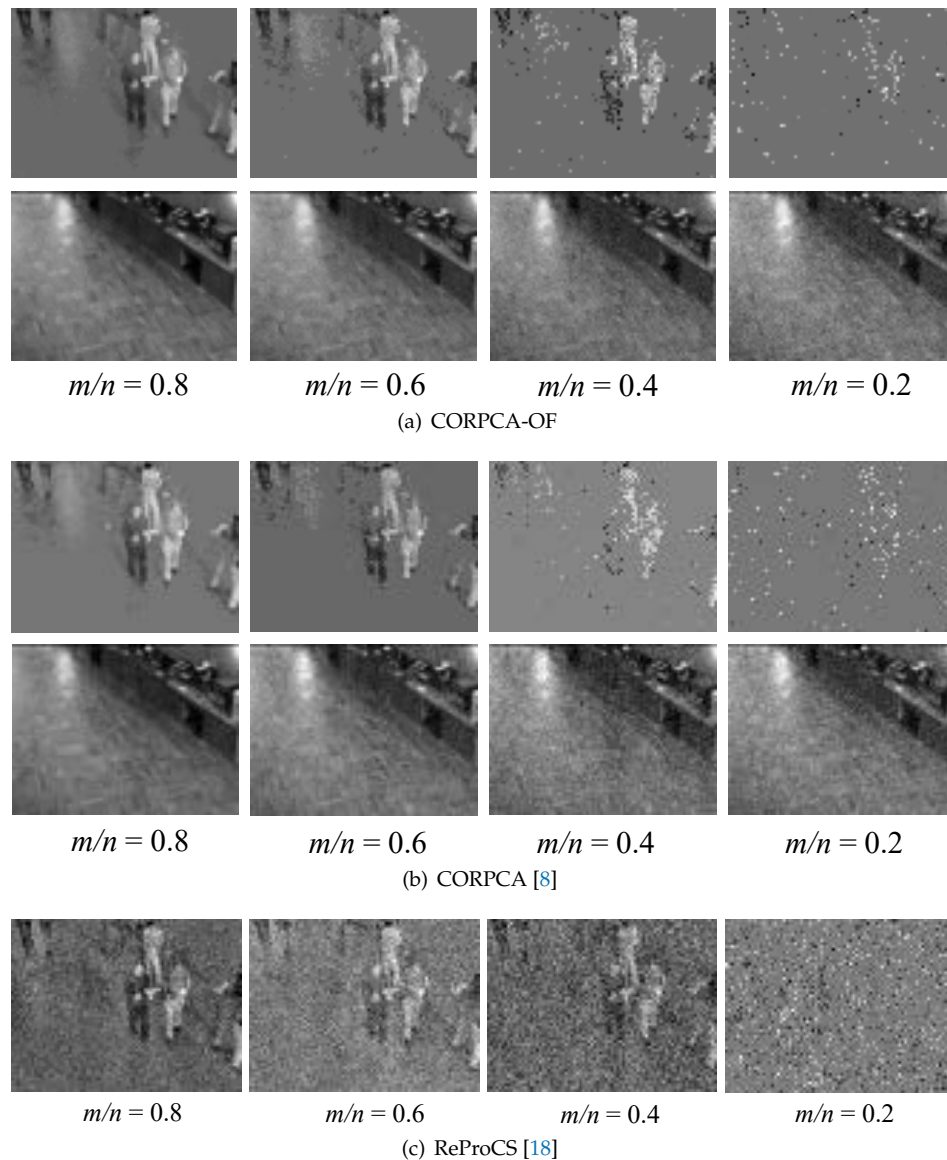


Figure 5. Compressive foreground and background separation of CORPCA-OF, CORPCA [8], and ReProCS [18] with different measurement rates m/n of frame Bootstrap #2213.

3.2.1. Visual Evaluation

We first consider background and foreground separation with full access to the video data; the visual results of the various methods are illustrated in Fig. 4. It is evident that, for both the video sequences, CORPCA-OF delivers superior visual results than the other methods, which suffer from less-details in the foreground and noisy background images. We can also observe improvements over CORPCA.

Additionally, we also compare the visual results of CORPCA-OF, CORPCA and ReProCS for the frames Bootstrap #2213 (in Fig. 5) and for Curtain #2866 (in Fig. 6) with compressed rates. They present the results under various rates on the number of measurements m over the dimension n of the data (the size of the vectorized frame) with rates: $m/n = \{0.8; 0.6; 0.4; 0.2\}$. Comparing CORPCA-OF with CORPCA, we can observe in Figs. 5 and 6 that CORPCA-OF gives the foregrounds that are less noisy and the background frames of higher visual quality. On comparison with ReProCS, our algorithm outperforms it significantly. At low rates, for instance with $m/n = 0.6$ (in Fig. 5(a)) or $m/n = 0.4$ (in Fig. 6(a)), the extracted foreground frames of CORPCA-OF are better than those of

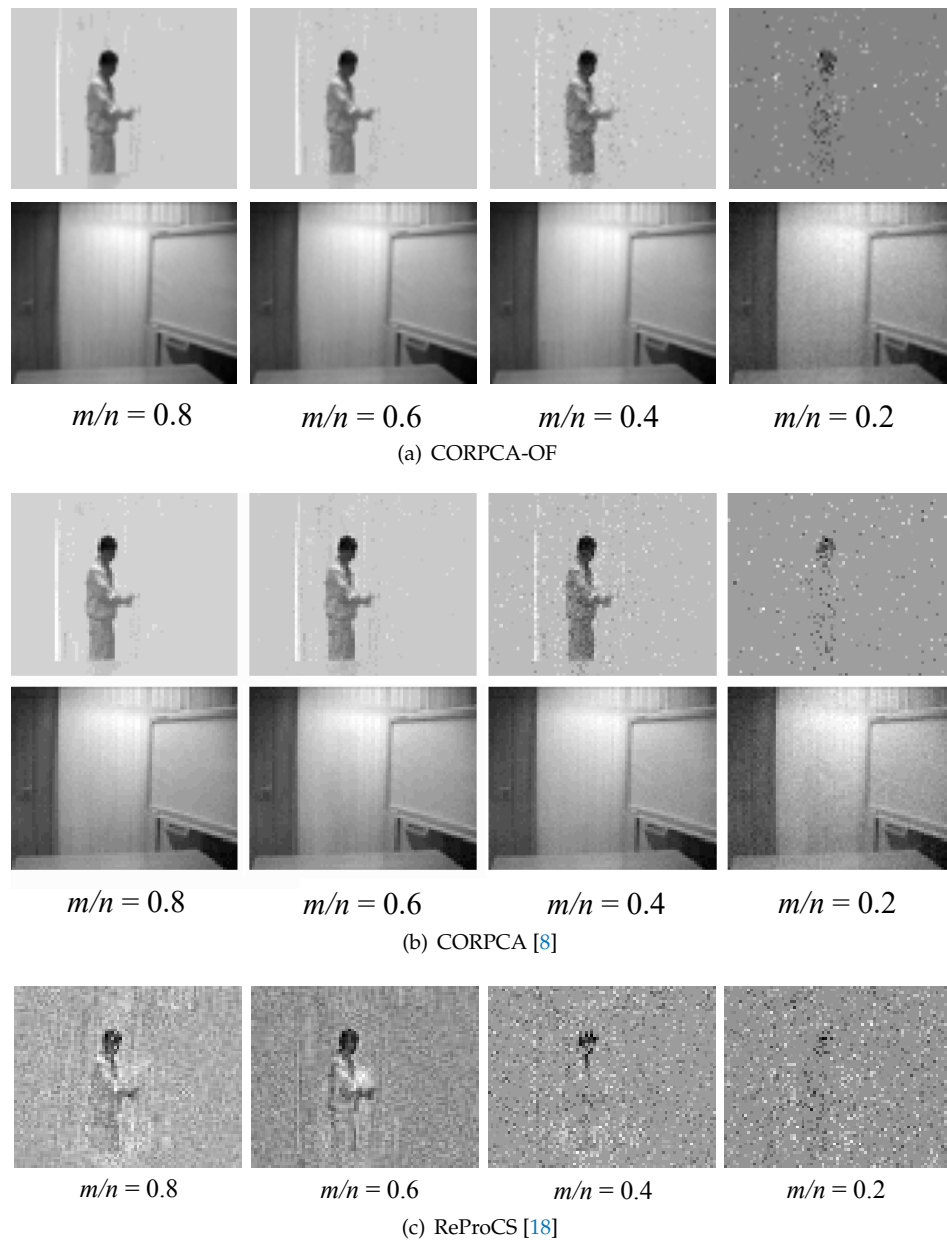


Figure 6. Compressive foreground and background separation of CORPCA-OF, CORPCA [8], and ReProCS [18] with different measurement rates m/n of frame Curtain #2866.

CORPCA and ReProCS. Even at a high rate of $m/n = 0.8$ the sparse components or the foreground frames using ReProCS are noisy and of poor visual quality. The Bootstrap sequence requires more measurements than Curtain due to the more complex foreground information. It is evident from Figs. 5 and 6 that the visual results obtained with CORPCA-OF are of superior quality compared to ReProCS and have significant improvements over CORPCA.

3.2.2. Quantitative Results

We evaluate quantitatively the separation performance via the *receiver operating curve* (ROC) metric [36]. The metrics *True positives* and *False positives* are defined in [36] as:

$$\text{True positives} = \frac{|\{\text{Foreground}\} \cap \{\text{Groundtruth Foreground}\}|}{|\{\text{Groundtruth Foreground}\}|} \quad (10)$$

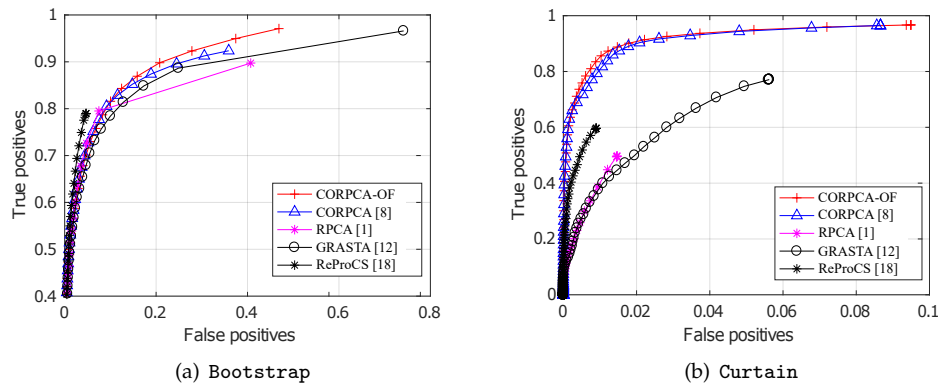
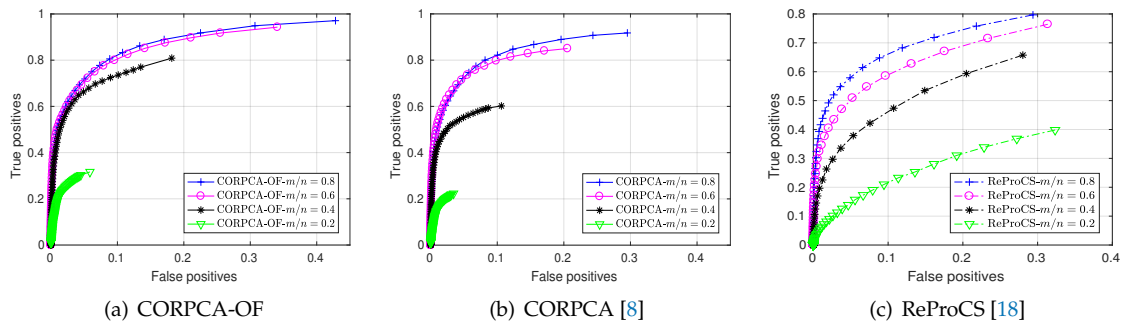
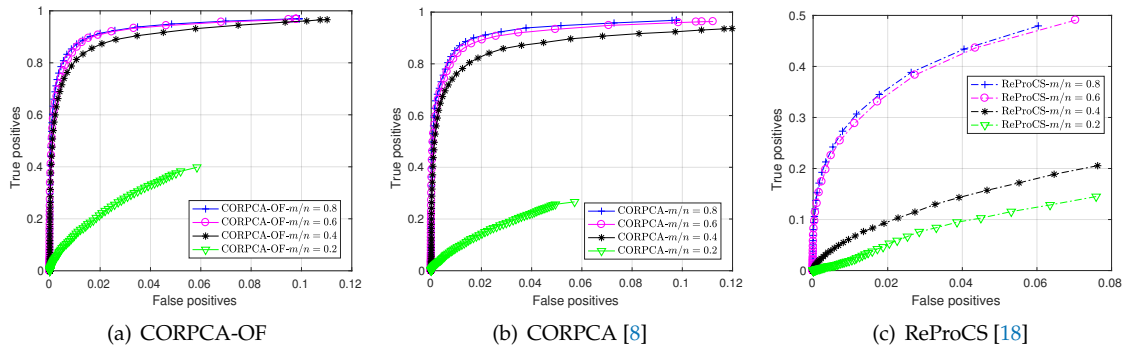


Figure 7. ROC for the different separation methods with full data.

Figure 8. ROC for CORPCA-OF, CORPCA [8], and ReProCS [18] with compressive measurement rates m/n for Bootstrap.Figure 9. ROC for CORPCA-OF, CORPCA [8], and ReProCS [18] with compressive measurement rates m/n for Curtain.

$$\text{False positives} = \frac{|\{\text{Foreground}\} \cap \{\text{Groundtruth Background}\}|}{|\{\text{Groundtruth Background}\}|} \quad (11)$$

Fig. 7 illustrates the ROC results when assuming full data access, i.e., $m/n = 1$, of CORPCA-OF, CORPCA, RPCA, GRASTA, and ReProCS. The results show that CORPCA-OF delivers higher performance than the other methods.

Furthermore, we compare the foreground recovery performance of CORPCA-OF against CORPCA and ReProCS for different compressive measurement rates: $m/n = \{0.8; 0.6; 0.4; 0.2\}$. The ROC results in Figs. 8 and 9 show that CORPCA-OF can achieve higher performance in comparison to ReProCS and CORPCA. In particular, with a small number of measurements, CORPCA-OF produces better curves

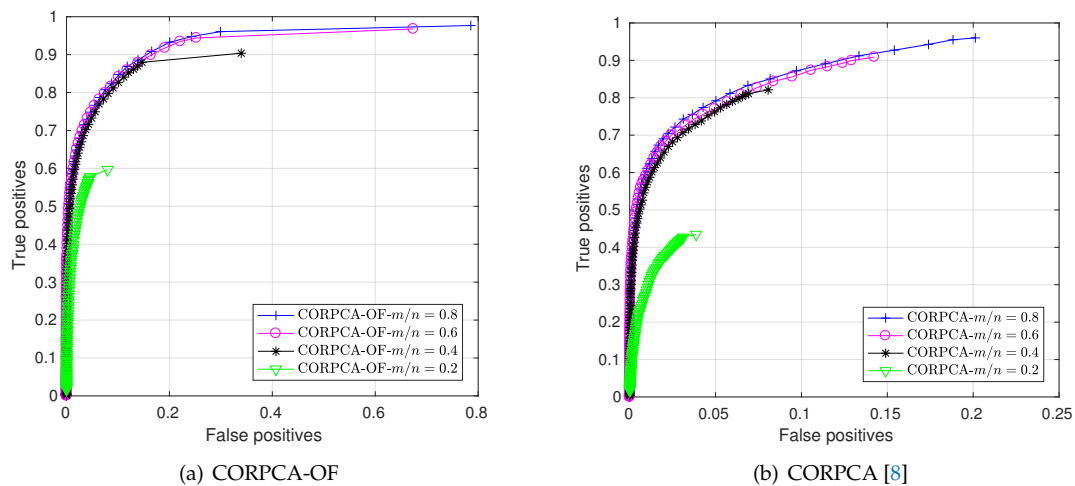


Figure 10. ROC of CORPCA-OF and CORPCA [8] methods with different rates for Escalator sequence.

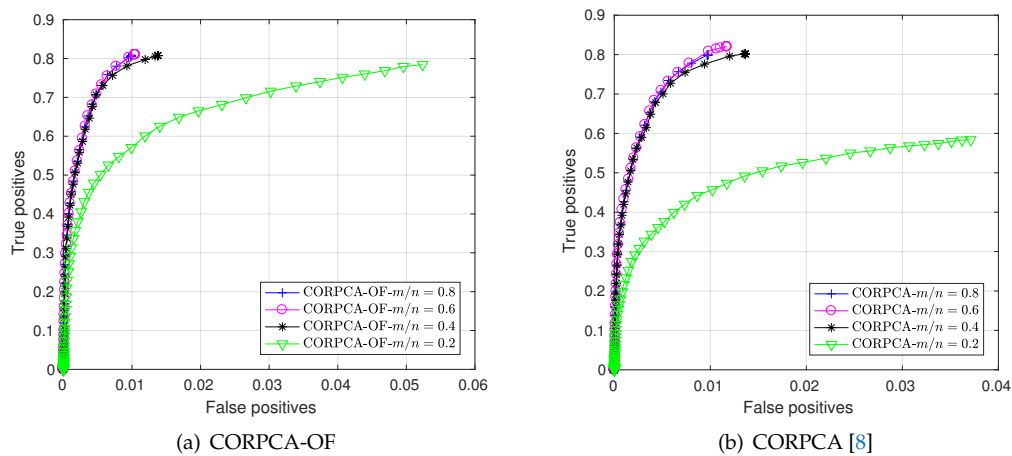


Figure 11. ROC of CORPCA-OF and CORPCA [8] methods with different rates for Fountain sequence.

than those of CORPCA. This is evident for Bootstrap at $m/n = \{0.2; 0.4; 0.6\}$ [see Fig. 8(a)]. For the Curtain sequence, which has a dynamic background and less complex foreground, the measurements at $m/n = \{0.2; 0.4\}$ [see Fig. 9(a)] are clearly better. The ROC results for ReProCS are quickly degraded even with a high compressive measurement rate $m/n = 0.8$ [see Figure 9(c)].

3.3. Additional Results

3.3.1. Escalator and Fountain sequences

The CORPCA-OF method was also compared with CORPCA against Escalator and Fountain sequences for compressive measurements. From Figs. 10(a) and 10(b), it is clear that CORPCA-OF performs slightly better than CORPCA. In Fig. 11(a) and Fig. 11(b) we can see that for the Fountain sequence, which is similar to the Curtain sequence in terms of complexity of foreground motions, the results are better for CORPCA-OF compared to CORPCA at rate $m/n = \{0.2\}$ and almost the same for higher rates.

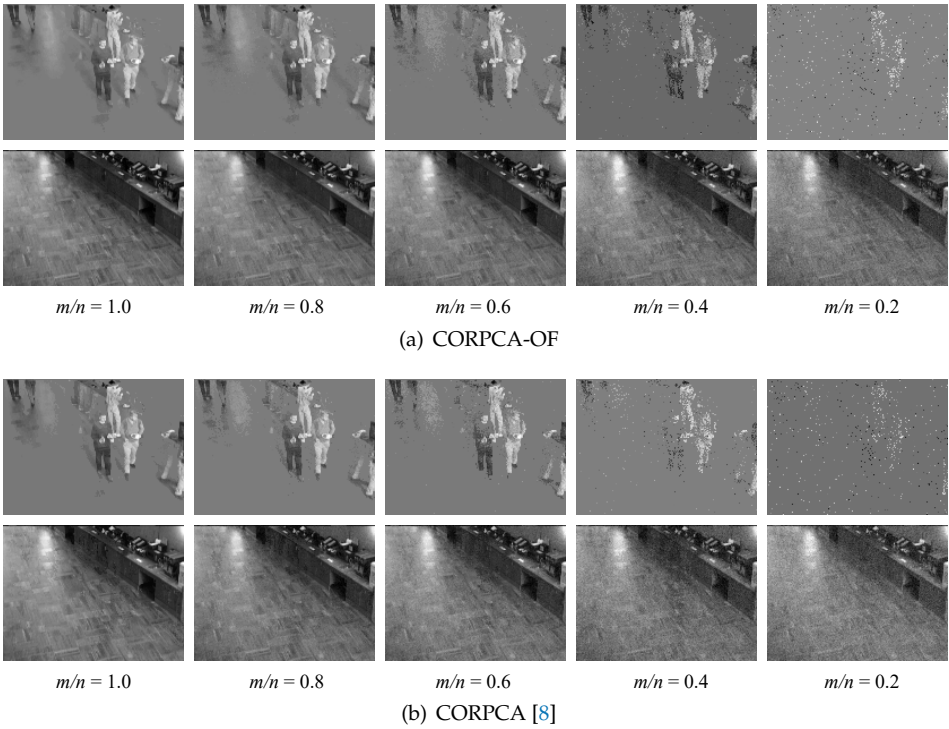


Figure 12. Comparison of CORPCA-OF and CORPCA [8] at full resolution with different rates m/n for Bootstrap #2213.

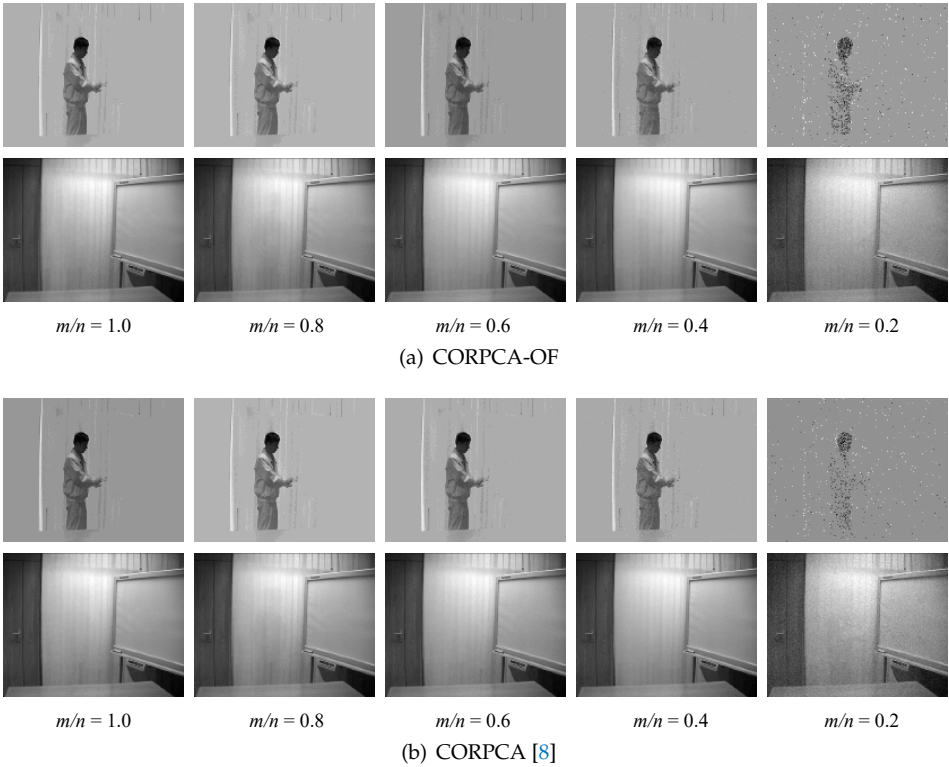


Figure 13. Comparison of CORPCA-OF and CORPCA [8] at full resolution with different rates m/n for Curtain #2866.

3.3.2. Visual Comparison of CORPCA-OF and CORPCA for Full Resolution

The visual comparison of CORPCA and CORPCA-OF for frame #2213 in full resolution (160×120) of the Bootstrap sequence can be seen in Fig.12. In Fig.13, we compare the full resolution (160×128) frame #2866 of the Curtain sequence. It can be seen that the background and foreground frames of CORPCA-OF for both Bootstrap and Curtain are much smoother and the contents have better structure compared to that of CORPCA. The improvements in foreground can be observed significantly at rates $m/n = \{0.6; 0.4; 0.2\}$ for the Bootstrap sequence for CORPCA-OF in Fig. 12(a) over CORPCA in Fig. 12(b). But in case of Curtain sequence, at low rates $m/n = \{0.4; 0.2\}$, there is significant improvement of foreground in Fig. 13(a) for CORPCA-OF over CORPCA in Fig. 13(b).

3.3.3. Separation Results with Various Datasets

HighwayI: This sequence is from the ATON dataset [37]. It is a sequence with fast motion of cars along a highway, with strong shadows and small camera jitter. The clear foreground and background separation for frames #285 to #293 can be seen from Fig.14.

Laboratory: This sequence is from the ATON dataset [37]. It contains an indoor sequence with two persons appearing for certain intervals. Original frames #142 to #158 are shown in Fig.15 along with the separated background and foreground frames.

The CORPCA-OF algorithm was also tested with other sequences namely, CMU dataset [38] (Fig. 16(a)), Airport (Fig. 16(b)), MovedObject (Fig. 16(c)) and WavingTrees (Fig. 16(d)) sequence from the wallflower dataset [39]. The results are as shown in Fig.16.

4. Conclusion

We have considered online video foreground background separation using RPCA approach from compressive measurements. A compressive online robust PCA algorithm with optical flow (CORPCA-OF) that can process one frame per time instance from compressive measurements was proposed. CORPCA-OF efficiently incorporates multiple prior frames based on the $n\text{-}\ell_1$ minimization problem. Furthermore, the proposed method can exploit motion estimation and compensation using optical flow to refine the prior information and obtain better quality separation. The proposed CORPCA-OF was tested on compressive online video separation application using numerous video sequences. The visual and quantitative results have shown the improvements on the prior generation and the superior performance offered by CORPCA-OF compared to the existing methods including the CORPCA baseline.

It is worth mentioning that CORPCA-OF can work in a general manner, with any number of prior frames. In this work, for the video separation, three prior frames are used as a compromise between performance and computational load. The C++ implementation of CORPCA-OF³ has been carried out, where the methods can be used as libraries for video separation. Building on this, the next step would be to make the algorithm more robust, adaptive, dynamic, and real-time by using GPUs.

Acknowledgments: This work was supported by the Chair of Multimedia Communications and Signal Processing at University of Erlangen-Nuremberg (FAU), Germany and the Humboldt Research Fellowship, the Alexander von Humboldt Foundation, Bonn, Germany.

Author Contributions: The work was carried out by Srivatsa Prativadibhayankaram based on the research and guidance of Huynh van Luong in collaboration with Thanh Ha Le and the supervision of Prof. André Kaup.

Conflicts of Interest: The authors declare no conflict of interest.

³ The code for the CORPCA-OF algorithm is available at <https://github.com/huynhnhlvd/corpc-af>

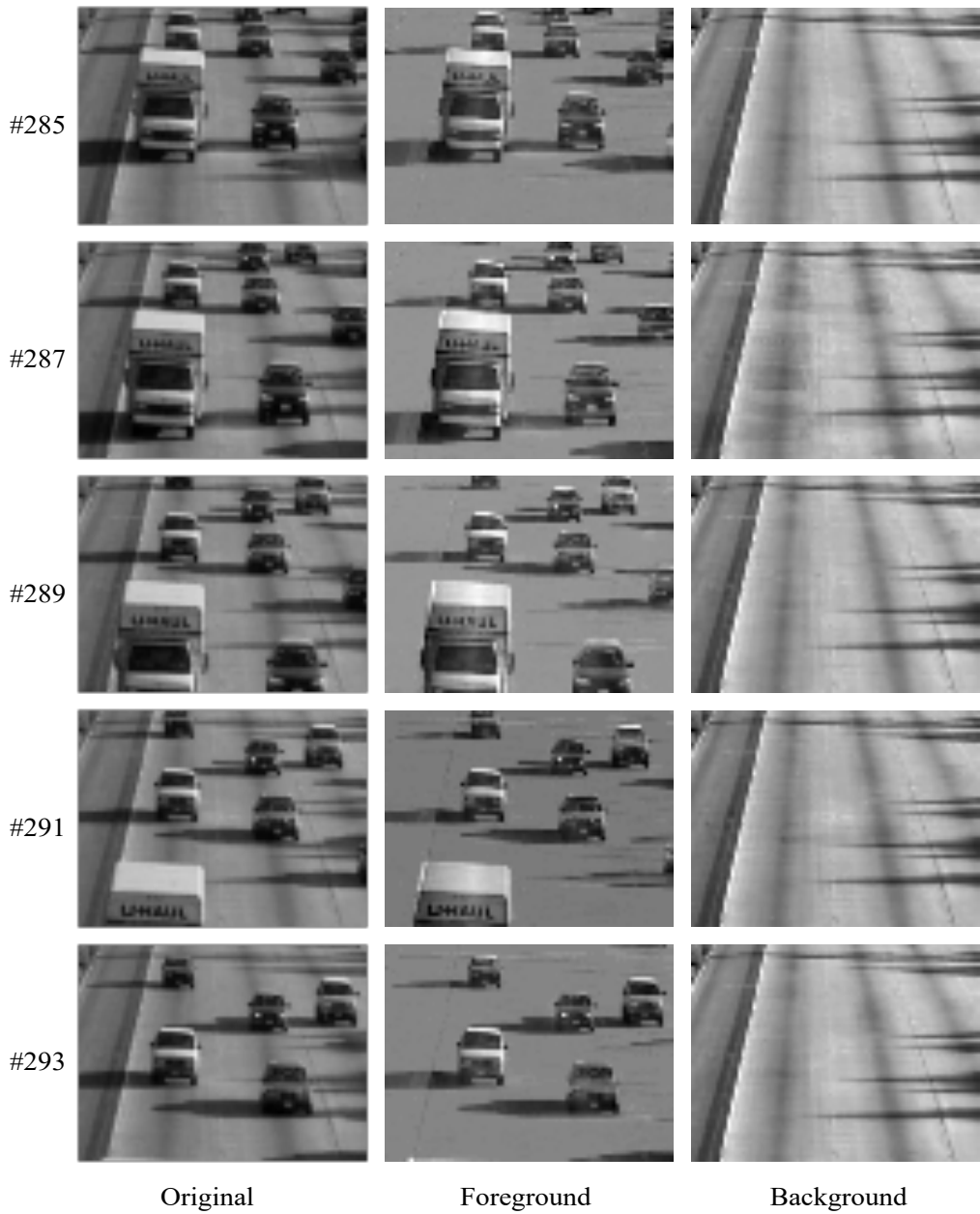


Figure 14. Foreground and background separation for HighwayI sequence.

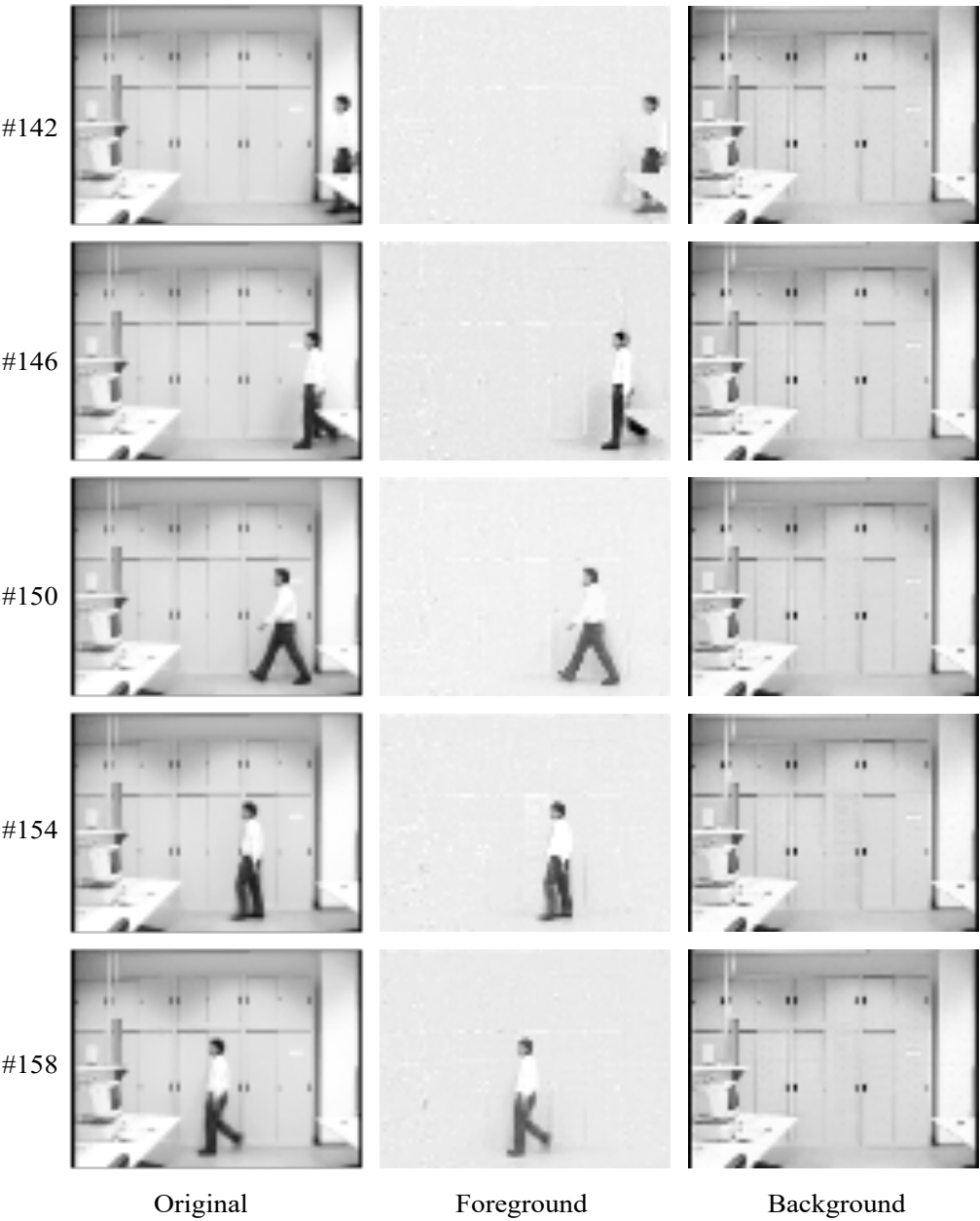


Figure 15. Foreground and background separation for Laboratory sequence.



Figure 16. Foreground and background separation of various sequences.

Abbreviations

The following abbreviations are used in this paper:

CORPCA	Compressive Online Robust Principal Component Analysis
CORPCA-OF	Compressive Online Robust Principal Component Analysis with Optical Flow
GPU	Graphical Processing Unit
GRASTA	Grassmannian Robust Adaptive Subspace Tracking Algorithm
OF	Optical Flow
PCA	Principal Component Analysis
PCP	Principal Component Pursuit
RAMSIA	Reconstruction Algorithm with Multiple Side Information using Adaptive weights
ReProCS	Recursive Projected Compressive Sensing
ROC	Receiver Operating Curve
RPCA	Robust Principal Component Analysis
SVD	Singular Value Decomposition

Appendix A. Overview of CORPCA-OF Implementaion in C++

An overview of the various C++ classes and functions implemented and a sample code snippet to demonstrate the usage of CORPCA-OF is as shown below.

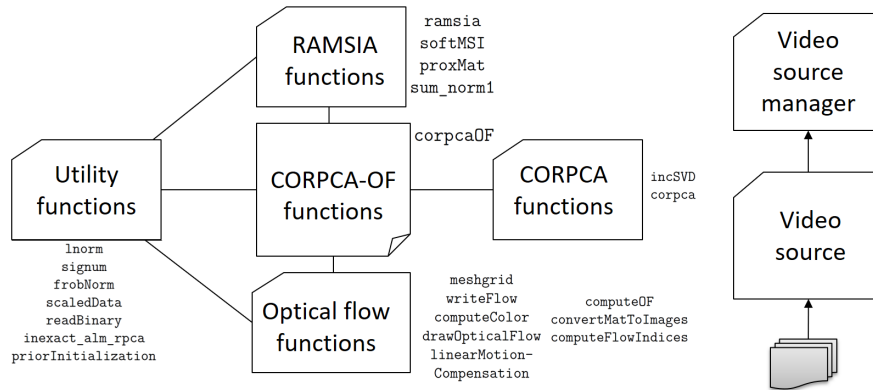


Figure A1. Overview of CORPCA-OF implementation

```
//initialize
VideoSourceManager* vsManager = VideoSourceManager::getInstance();
vsManager->initialize();
vsManager->setDataPath("$dataPath");
vsManager->setOutDataPath("$outImgDir");
//choose video
vsManager->selectVideo(VideoSourceManager::BOOTSTRAP);
//set parameters for video source
VideoSource* vSource = vsManager->getVideoSource();
vSource->setScale(0.5);
vSource->setRate(1.0);
cv::Mat fgCorpcaOF, bgCorpcaOF;
//start CORPCA-OF
corpcaOF(vsManager, fgCorpcaOF, bgCorpcaOF);
//save the results
vsManager->saveCorpcaOFOutput(fgCorpcaOF, bgCorpcaOF);
```

Figure A2. A sample of CORPCA-OF code usage

References

1. Candès, E.J.; Li, X.; Ma, Y.; Wright, J. Robust principal component analysis? *Journal of the ACM (JACM)* **2011**, *58*, 11.
2. Wright, J.; Ganesh, A.; Rao, S.; Peng, Y.; Ma, Y. Robust principal component analysis: Exact recovery of corrupted low-rank matrices via convex optimization. *Advances in neural information processing systems*, **2009**, pp. 2080–2088.
3. Horn, B.K.; Schunck, B.G. Determining optical flow. *Artificial intelligence* **1981**, *17*, 185–203.
4. Bruhn, A.; Weickert, J.; Schnörr, C. Lucas/Kanade meets Horn/Schunck: Combining local and global optic flow methods. *International journal of computer vision* **2005**, *61*, 211–231.
5. Baker, S.; Scharstein, D.; Lewis, J.; Roth, S.; Black, M.J.; Szeliski, R. A database and evaluation methodology for optical flow. *International Journal of Computer Vision* **2011**, *92*, 1–31.
6. Brox, T.; Malik, J. Large displacement optical flow: descriptor matching in variational motion estimation. *IEEE transactions on pattern analysis and machine intelligence* **2011**, *33*, 500–513.

7. Prativadibhayankaram, S.; Luong, H. V.; Le, T.H.; Kaup, A. Compressive Online Robust Principal Component Analysis with Optical Flow for Video Foreground-Background Separation. *Proceedings of the Eighth International Symposium on Information and Communication Technology*. ACM, **2017**, pp.385–392.
8. Luong, H. V.; Deligiannis, N.; Seiler, J.; Forchhammer, S.; Kaup, A. Compressive online robust principal component analysis with multiple prior information. *IEEE Global Conference on Signal and Information Processing*. Montreal, Canada, **2017**.
9. Rodriguez, P.; Wohlberg, B. Incremental principal component pursuit for video background modeling. *Journal of Mathematical Imaging and Vision* **2016**, *55*, 1–18.
10. Hong, B.; Wei, L.; Hu, Y.; Cai, D.; He, X. Online robust principal component analysis via truncated nuclear norm regularization. *Neurocomputing* **2016**, *175*, 216–222.
11. Xiao, W.; Huang, X.; Silva, J.; Emrani, S.; Chaudhuri, A. Online Robust Principal Component Analysis with Change Point Detection. *arXiv preprint arXiv:1702.05698* **2017**.
12. He, J.; Balzano, L.; Szlam, A. Incremental gradient on the grassmannian for online foreground and background separation in subsampled video. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, **2012**, pp. 1568–1575.
13. Xu, J.; Ithapu, V.K.; Mukherjee, L.; Rehg, J.M.; Singh, V. Gosus: Grassmannian online subspace updates with structured-sparsity. *IEEE International Conference on Computer Vision (ICCV)*, **2013**, pp.3376–3383.
14. Feng, J.; Xu, H.; Yan, S. Online robust pca via stochastic optimization. *Advances in Neural Information Processing Systems*, **2013**, pp. 404–412.
15. Mansour, H.; Jiang, X. A robust online subspace estimation and tracking algorithm. *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, **2015**, pp. 4065–4069.
16. Wright, J.; Ganesh, A.; Min, K.; Ma, Y. Compressive principal component pursuit. *Information and Inference: A Journal of the IMA* **2013**, *2*, 32–68.
17. Pan, P.; Feng, J.; Chen, L.; Yang, Y. Online compressed robust PCA. *International Joint Conference on Neural Networks (IJCNN)*, **2017**, pp. 1041–1048.
18. Guo, H.; Qiu, C.; Vaswani, N. An online algorithm for separating sparse and low-dimensional signal sequences from their sum. *IEEE Transactions on Signal Processing* **2014**, *62*, 4284–4297.
19. Mota, J.F.; Deligiannis, N.; Sankaranarayanan, A.C.; Cevher, V.; Rodrigues, M.R. Adaptive-rate reconstruction of time-varying signals with application in compressive foreground extraction. *IEEE Transactions on Signal Processing* **2016**, *64*, 3651–3666.
20. Warnell, G.; Bhattacharya, S.; Chellappa, R.; Basar, T. Adaptive-Rate Compressive Sensing Using Side Information. *IEEE Transactions on Image Processing* **2015**, *24*, 3846–3857.
21. Qiu, C.; Vaswani, N.; Lois, B.; Hogben, L. Recursive Robust PCA or Recursive Sparse Recovery in Large but Structured Noise. *IEEE Transactions on Information Theory* **2014**, *60*, 5007–5039.
22. Zhan, J.; Vaswani, N. Robust PCA with partial subspace knowledge. *IEEE International Symposium on Information Theory*, **2014**.
23. Vaswani, N.; Zhan, J. Recursive Recovery of Sparse Signal Sequences From Compressive Measurements: A Review. *IEEE Transactions on Signal Processing* **2016**, *64*, 3523–3549.
24. Bouwmans, T.; Sobral, A.; Javed, S.; Jung, S.K.; Zahzah, E.H. Decomposition into Low-rank Plus Additive Matrices for Background/Foreground Separation. *Comput. Sci. Rev.* **2017**, *23*, 1–71.
25. Vaswani, N.; Lu, W. Modified-CS: Modifying compressive sensing for problems with partially known support. *IEEE Transactions on Signal Processing* **2010**, *58*, 4595–4607.
26. Sobral, A.; Bouwmans, T.; ZahZah, E.h. Double-constrained RPCA based on saliency maps for foreground detection in automated maritime surveillance. *12th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, **2015**, pp. 1–6.
27. Javed, S.; Mahmood, A.; Bouwmans, T.; Jung, S.K. Background-Foreground Modeling Based on Spatiotemporal Sparse Subspace Clustering. *IEEE Transactions on Image Processing* **2017**, *26*, 5840–5854.
28. Javed, S.; Mahmood, A.; Bouwmans, T.; Jung, S.K. Spatiotemporal low-rank modeling for complex scene background initialization. *IEEE Transactions on Circuits and Systems for Video Technology* **2016**.
29. Berjón, D.; Cuevas, C.; Morán, F.; García, N. Real-time nonparametric background subtraction with tracking-based foreground update. *Pattern Recognition* **2018**, *74*, 156–170.
30. Brand, M. Incremental Singular Value Decomposition of Uncertain Data with Missing Values. *European Conference on Computer Vision*, **2002**.

31. Luong, H. V.; Seiler, J.; Kaup, A.; Forchhammer, S. Sparse Signal Reconstruction with Multiple Side Information using Adaptive Weights for Multiview Sources. *IEEE International Conference on Image Processing*, **2016**.
32. Beck, A.; Teboulle, M. A Fast Iterative Shrinkage-Thresholding Algorithm for Linear Inverse Problems. *SIAM Journal on Imaging Sciences* **2009**, *2*(1), 183–202.
33. Cai, J.F.; Candès, E.J.; Shen, Z. A Singular Value Thresholding Algorithm for Matrix Completion. *SIAM J. on Optimization* **2010**, *20*, 1956–1982.
34. Szeliski, R. In *Computer vision: algorithms and applications*; Springer Science & Business Media, 2010.
35. Li, L.; Huang, W.; Gu, I.Y.H.; Tian, Q. Statistical modeling of complex backgrounds for foreground object detection. *IEEE Transactions on Image Processing* **2004**, *13*, 1459–1472.
36. Dikmen, M.; Tsai, S.F.; Huang, T.S. Base selection in estimating sparse foreground in video. *16th IEEE International Conference on Image Processing (ICIP)*, **2009**, pp. 3217–3220.
37. Cucchiara, R.; Grana, C.; Piccardi, M.; Prati, A. Detecting moving objects, ghosts, and shadows in video streams. *IEEE transactions on pattern analysis and machine intelligence* **2003**, *25*, 1337–1342.
38. Sheikh, Y.; Shah, M. Bayesian modeling of dynamic scenes for object detection. *IEEE transactions on pattern analysis and machine intelligence* **2005**, *27*, 1778–1792.
39. Toyama, K.; Krumm, J.; Brumitt, B.; Meyers, B. Wallflower: principles and practice of background maintenance. *Proceedings of the Seventh IEEE International Conference on Computer Vision*, **1999**, Vol. 1, pp. 255–261.