Journal of
Imaging

MDPI

Article

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

Srivatsa Prativadibhayankaram [1,†], **Huynh Van Luong** [1,*,‡], **Thanh Ha Le** [2] and **André Kaup** [1]

[1] Multimedia Communications and Signal Processing, University of Erlangen-Nuremberg, 91058 Erlangen, Germany; srivatsa.pv@live.com (S.P.); andre.kaup@fau.de (A.K.)

[2] Human Machine Interaction, University of Engineering and Technology, Vietnam National University, Hanoi 100000, Vietnam; ltha@vnu.edu.vn

[*] Correspondence: hvanluon@etrovub.be; Tel.: +32-2-629-1674; Fax: +32-2-629-2883

[†] Current address: Digital Cinema Group, Fraunhofer IIS, 91058 Erlangen, Germany.

[‡] Current address: Department of Electronics and Informatics, Vrije Universiteit Brussel, 1050 Brussels, Belgium.

Received: 1 May 2018; Accepted: 27 June 2018; Published: 3 July 2018

check for updates

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 operates on a small set of measurements taken per frame, 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-ℓ_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 most important components for object detection, identification, and tracking. In video separation, a video sequence can be separated into a slowly changing background (modeled by $\boldsymbol{L}$ as a low-rank component) and the foreground (modeled by $\boldsymbol{S}$, 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 $\boldsymbol{M}$ into the sum of unknown sparse $\boldsymbol{S}$ and low-rank $\boldsymbol{L}$ by solving the Principal Component Pursuit (PCP) [1] problem:

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

where $\|\cdot\|_*$ is the matrix nuclear norm (sum of singular values) and $\|\cdot\|_1$ is the ℓ_1-norm (sum of absolute values). 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 $\boldsymbol{M}$) 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 $\boldsymbol{M}_t$ into $\boldsymbol{S}_t = [\boldsymbol{x}_1\ \boldsymbol{x}_2\ ...\ \boldsymbol{x}_t]$ 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 $\boldsymbol{S}_t$ and $\boldsymbol{L}_t$, respectively. We assume that $\boldsymbol{S}_{t-1} = [\boldsymbol{x}_1\ \boldsymbol{x}_2\ ...\ \boldsymbol{x}_{t-1}]$ and $\boldsymbol{L}_{t-1} = [\boldsymbol{v}_1\ \boldsymbol{v}_2\ ...\ \boldsymbol{v}_{t-1}]$ 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 $\boldsymbol{x}_t + \boldsymbol{v}_t$, that is, we observe $\boldsymbol{y}_t = \boldsymbol{\Phi}(\boldsymbol{x}_t + \boldsymbol{v}_t)$, where $\boldsymbol{\Phi} \in \mathbb{R}^{m \times n} (m < n)$ is a random projection, $\boldsymbol{x}_t$ is the sparse component (foreground) and $\boldsymbol{v}_t$ is the low-rank component (background) at time instance t. We proceed with the assumption that video can be seperated into dynamic, sparse foreground components and a static or slowly changing, low-rank background components. The recovery problem at time instance t is thus written [7] as

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

where $\boldsymbol{L}_{t-1}$ and $\boldsymbol{S}_{t-1}$ are given.

1.1. Related Work

Many methods [8–14] have been proposed to solve the separation problem by advancing RPCA [1]. Incremental PCP [8] processes each column-vector in $\boldsymbol{M}$ 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 [15]. The studies in [11–14,16] 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 $\boldsymbol{x}_t$ in (2) has been proposed in [17], however, the low-rank component $\boldsymbol{v}_t$ in (2) is not recovered per time instance from a small number of measurements.

The alternative method in [18,19] estimates the number of compressive measurements required to recover foreground $\boldsymbol{x}_t$ per time instance via assuming the background $\boldsymbol{v}_t$ not-varying. This assumption is invalid in realistic scenarios due to variations in illumination or dynamic backgrounds. The online method in [16] works on compressive measurements without taking the prior information into account.

Separating a video sequence or a set of frames using prior information brings about significant improvements in the context of online RPCA [17,20,21]. Some studies on recursive recovery from low-dimensional measurements have been proposed to leverage prior information [17,18,20,22]. The study in [22] provided a comprehensive overview of the domain, reviewing a class of recursive algorithms. A review of all the recent problem formulations is made in [23] with a unified view with Decomposition into Low-rank plus Additive Matrices (DLAM).

The studies in [17,20] used modified-CS [24] to leverage prior knowledge under the condition of slowly varying support and signal values. However, this method as well as the methods in [11,12,14] do not explore the correlations between the current frame and multiple previously separated frames. The recent work in [7] 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. An interesting method proposed in [25] makes use of the low-rank and sparse components of a video frame in addition to a dense noise component ($\boldsymbol{G}$), i.e., $\boldsymbol{M} = \boldsymbol{L} + \boldsymbol{S} + \boldsymbol{G}$ for exoplanet detection.

There are other 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.

Background subtraction is a widely used technique to separate foreground and background. A detailed review of commonly used techniques in performing background subtraction has been made in [30]. A survey of all the existing methods in background subtraction has been thoroughly carried out in [31] and another survey [32] is focused on detecting stationary foreground objects. Evaluation of 29 background subtraction methods using the BMC dataset [33] has been performed in [34]. An interesting technique by making use of genetic programming can be found in [35]. This is a powerful method that uses a repertoire of different background subtraction algorithms and chooses a method or a combination of methods appropriately, based on the video sequence in order to obtain better results. Semantics based approach is proposed in [36]. This method improves a background separation algorithm by making use of semantics to identify objects as foregound or background. It is worth noting that, in this method, semantics are used as prior information. In [37], a method is proposed to analyze dynamic background region and reduce the false positives by checking the false positives again. If foreground is detected in the dynamic background region, it is removed by re-checking false positives from the dynamic background samples. Ref. [38] makes use of a different approach to perform background subtraction. This method evaluates the importance of each background sample in an online manner based on the recurrence among local all the local observations. A persistence based word dictionary is used, which addresses both short term and long term adaptation at pixel level and frame level. A good set of videos to test any background separation algorithm are provided in [33,39]. Ref. [33] is called the BMC (Background Models Challenge) and presents a benchmark dataset and evaluation process built from both synthetic and real videos. It is focused on outdoor sequences and with varying weather conditions. Additionally, an evaluation criterion is also provided with an associated software. CDnet dataset in [33,39] focuses on evaluation of change and motion detection approaches. It contains challenging scenarios to test background subtraction and motion detection.

Another closely related field is background generation. Having a background image free of any foreground objects is important in most applications. Generating a stationary background image is challenging when the background is not fully visible. And this method needs a finite data volume but it is specific for each scene. A detailed survey focused on model initialization for background is carried out in [40]. It provides a basis for easy comparison of existing and new methods using a common set of ground truth sequences. Another survey and benchmarking of scene background initialization methods has been extensively carried out in [41] using several evaluation metrics on a large video dataset called SBMNet [42]. Background generation using motion detection is proposed in [43]. It uses a temporal median filter and a patch selection mechanism based on motion detection performed by a background subtraction algorithm. An enhanced version of this method is made in [44] by using optical flow for motion detection. It leverages memoryless dense optical flow algorithms to compute velocity vector between two frames for each pixel. There are methods that make use of neural networks. One such method is described in [45], which performs background estimation using weightless neural networks. This algorithm could be used as a preliminary step in video foreground detection, as detecting foreground or moving objects is not in the scope of this method.

1.2. Contributions

In this paper, we propose a compressive online robust PCA with optical flow (CORPCA-OF) method (Our work has been reported in [46]), which is based on our previous work in [7]. We leverage information from previously separated foreground frames via optical flow [6]. The novelty of CORPCA-OF over CORPCA [7] 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 [47] method. The compressive separation problem in (2) is solved in an online manner by minimizing not only an n-ℓ_1-norm cost function [48] 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 [7], on which our proposed CORPCA-OF is built upon. Based on this, we formulate the problem statement in Section 2.1, followed by the proposed CORPCA-OF algorithm in Section 2.2.2. The visual and quantitative results obtained by testing our method on real video sequences are presented and discussed in Section 3.

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

In this section, we firstly review the CORPCA [7] algorithm for online compressive video separation and state our problem. Thereafter, we propose our CORPCA-OF method, which is summarized in the CORPCA-OF algorithm.

2.1. Compressive Online Robust PCA (CORPCA) for Video Separation

The CORPCA algorithm [7] for video separation is based on Reconstruction Algorithm with Multiple Side Information using Adaptive weights (RAMSIA) [48], that solves an n-ℓ_1 minimization problem with adaptive weights to recover a sparse signal $\boldsymbol{x}$ from low-dimensional random measurements $\boldsymbol{y} = \boldsymbol{\Phi x}$ with the aid of multiple prior information or side information $\boldsymbol{z}_j$, $j \in \{0, 1, \ldots, J\}$, with $\boldsymbol{z}_0 = \boldsymbol{0}$.The objective function of RAMSIA [48] is given by

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

where $\lambda > 0$ is a regularization parameter 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 $\boldsymbol{z}_j$; namely, $\mathbf{W}_j = \mathrm{diag}(w_{j1}, w_{j2}, ..., w_{jn})$ with $w_{ji} > 0$ being the weight for the i-th element in the $\boldsymbol{z}_j$ vector.

The 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 $\boldsymbol{y}_t = \boldsymbol{\Phi}(\boldsymbol{x}_t + \boldsymbol{v}_t)$ with $\boldsymbol{y}_t \in \mathbb{R}^m$. Let $\boldsymbol{Z}_{t-1} := \{\boldsymbol{z}_1, ..., \boldsymbol{z}_J\}$, a set of $\boldsymbol{z}_j \in \mathbb{R}^n$, and $\boldsymbol{B}_{t-1} \in \mathbb{R}^{n \times d}$ denote prior information for $\boldsymbol{x}_t$ and $\boldsymbol{v}_t$, respectively. The prior information $\boldsymbol{Z}_{t-1}$ and $\boldsymbol{B}_{t-1}$ are formed by using the already reconstructed set of vectors $\{\hat{\boldsymbol{x}}_1, ..., \hat{\boldsymbol{x}}_{t-1}\}$ and $\{\hat{\boldsymbol{v}}_1, ..., \hat{\boldsymbol{v}}_{t-1}\}$.

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

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

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

The CORPCA algorithm (The source code, the test sequences, and the corresponding outcomes of CORPCA are available at [49]) solves Problem (4) given that $\boldsymbol{Z}_{t-1}$ and $\boldsymbol{B}_{t-1}$ are known (they are obtained from the time instance or recursion). Thereafter, we update $\boldsymbol{Z}_t$ and $\boldsymbol{B}_t$, which are used in the subsequent time instance.

Let us denote $f(\boldsymbol{v}_t,\boldsymbol{x}_t) = (1/2)\|\boldsymbol{\Phi}(\boldsymbol{x}_t+\boldsymbol{v}_t)-\boldsymbol{y}_t\|_2^2$, $g(\boldsymbol{x}_t) = \lambda\sum_{j=0}^{J}\beta_j\|\mathbf{W}_j(\boldsymbol{x}_t-\boldsymbol{z}_j)\|_1$, and $h(\boldsymbol{v}_t)=\|[\boldsymbol{B}_{t-1}\ \boldsymbol{v}_t]\|_*$, where $f(.)$ is a function of both background and foreground, $g(.)$ is a function of the foreground (sparse) component and $h(.)$ is a function of the background (low rank) component. The components $\boldsymbol{x}_t^{(k+1)}$ and $\boldsymbol{v}_t^{(k+1)}$ are solved iteratively at iteration $k+1$ via the soft thresholding operator [50] for $\boldsymbol{x}_t$ and the single value thresholding operator [51] for $\boldsymbol{v}_t$:

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

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

2.2. Video Foreground and Background Separation Using CORPCA-OF

Problem statement: Using the prior information in CORPCA [7] has provided significant improvement of the current frame separation. However, there can be displacements between the consecutive frames that can deteriorate the separation performance. Figure 1 illustrates an example of three previous foreground frames, $\boldsymbol{x}_{t-3}, \boldsymbol{x}_{t-2}$, and $\boldsymbol{x}_{t-1}$. These frames can be used directly as prior information to recover foreground $\boldsymbol{x}_t$ and background $\boldsymbol{v}_t$ due to temporal correlations between $\boldsymbol{x}_t$ and $\boldsymbol{x}_{t-3}, \boldsymbol{x}_{t-2}, \boldsymbol{x}_{t-1}$, as seen in CORPCA. In the last row of the prior foreground frames in Figure 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 Figure 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 $\boldsymbol{x}'_{t-3}, \boldsymbol{x}'_{t-2}$ with $\boldsymbol{x}_{t-3}, \boldsymbol{x}_{t-2}, \boldsymbol{x}_{t-1}$), and it is better correlated to $\boldsymbol{x}_t$. In this work, we discuss a new algorithm—CORPCA with Optical Flow (CORPCA-OF), whose work flow is shown in Figure 1. Optical flow [6] is used to improve prior foreground frames.

2.2.1. Compressive Separation Model with CORPCA-OF

A compressive separation model using the CORPCA-OF method is shown in Figure 2. At a time instance t, the inputs consist of compressive measurements $\boldsymbol{y}_t = \boldsymbol{\Phi}(\boldsymbol{x}_t+\boldsymbol{v}_t)$ and prior information from time instance $t-1$, $\boldsymbol{Z}_{t-1}$ (foreground) and $\boldsymbol{B}_{t-1}$ (background). The model outputs foreground and background information $\boldsymbol{x}_t$ and $\boldsymbol{v}_t$ by solving the CORPCA minimization problem in (4). Finally, the outputs $\boldsymbol{x}_t$ and $\boldsymbol{v}_t$ are used to generate better prior foreground information via a prior generation using optical flow and update $\boldsymbol{Z}_{t-1}$ and $\boldsymbol{B}_{t-1}$ for the next instance via a prior update. The novel block of CORPCA-OF compared with CORPCA [7] 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.

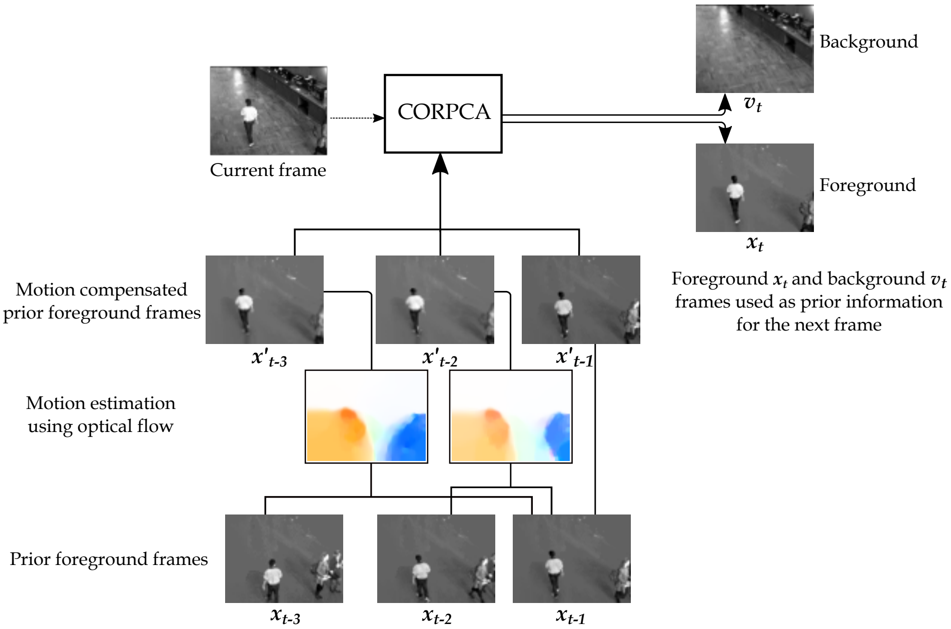

Background
v_t
CORPCA
Current frame
Foreground
x_t
Foreground x_t and background v_t frames used as prior information for the next frame
Motion compensated prior foreground frames
x'_{t-3}
x'_{t-2}
x'_{t-1}
Motion estimation using optical flow
Prior foreground frames
x_{t-3}
x_{t-2}
x_{t-1}

Figure 1. CORPCA-OF work flow.

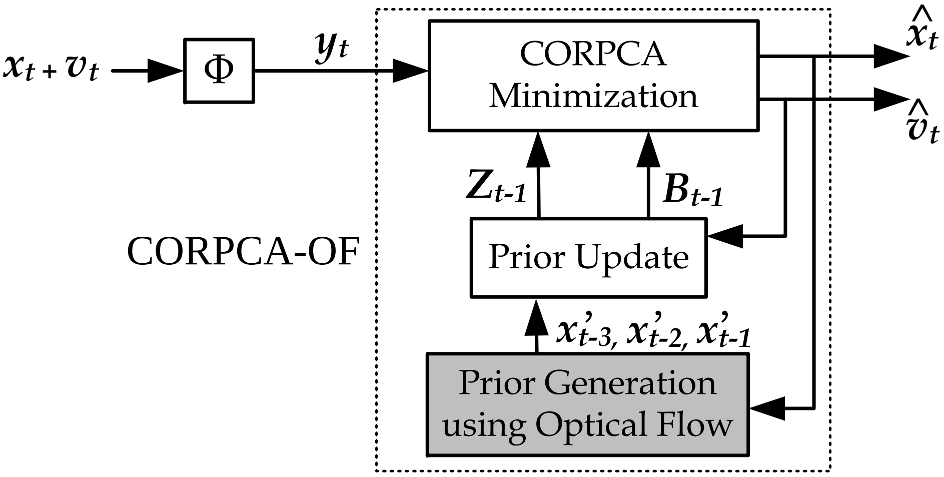


Figure 2. Compressive separation model using CORPCA-OF .

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 Figure 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 $\mathbf{v}_{1x}, \mathbf{v}_{2x}$ and $\mathbf{v}_{1y}, \mathbf{v}_{2y} \in \mathbb{R}^n$, respectively. The estimated motions in the form of optical flow vectors, $(\mathbf{v}_{1x}, \mathbf{v}_{1y})$ and $(\mathbf{v}_{2x}, \mathbf{v}_{2y})$, 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.

Algorithm 1: The proposed CORPCA-OF algorithm.

```
   Input: $\boldsymbol{y}_t, \boldsymbol{Z}_{t-1}, \boldsymbol{B}_{t-1}$;
   Output: $\widehat{\boldsymbol{x}}_t, \widehat{\boldsymbol{v}}_t, \boldsymbol{Z}_t, \boldsymbol{B}_t$;
   // Initialize variables and parameters.
 1 $\boldsymbol{x}_t^{(-1)}=\boldsymbol{x}_t^{(0)}=\mathbf{0}; \boldsymbol{v}_t^{(-1)}=\boldsymbol{v}_t^{(0)}=\mathbf{0}; \xi_{-1}=\xi_0=1; \mu_0=0; \bar{\mu}>0; \lambda>0; 0<\epsilon<1; k=0; g_1(\cdot)=\|\cdot\|_1$;
   // Motion estimation and compensation using Large Displacement Optical
      Flow [6]
 2 $(\mathbf{v}_{1\mathrm{x}}, \mathbf{v}_{1\mathrm{y}}) = f_{ME}(\boldsymbol{x}_{t-1}, \boldsymbol{x}_{t-2})$;
 3 $(\mathbf{v}_{2\mathrm{x}}, \mathbf{v}_{2\mathrm{y}}) = f_{ME}(\boldsymbol{x}_{t-1}, \boldsymbol{x}_{t-3})$;
 4 $\boldsymbol{x}'_{t-1} = \boldsymbol{x}_{t-1}$;
 5 $\boldsymbol{x}'_{t-2} = f_{MC}(\boldsymbol{x}_{t-1}, \mathbf{v}_{1\mathrm{x}}, \mathbf{v}_{1\mathrm{y}})$;
 6 $\boldsymbol{x}'_{t-3} = f_{MC}(\boldsymbol{x}_{t-1}, \frac{1}{2}\mathbf{v}_{2\mathrm{x}}, \frac{1}{2}\mathbf{v}_{2\mathrm{y}})$;
 7 $\boldsymbol{z}_J = \boldsymbol{x}'_{t-1}; \boldsymbol{z}_{J-1} = \boldsymbol{x}'_{t-2}; \boldsymbol{z}_{J-2} = \boldsymbol{x}'_{t-3}$;
 8 while not converged do
       // Solve Problem (4).
 9     $\widetilde{\boldsymbol{v}}_t^{(k)} = \boldsymbol{v}_t^{(k)} + \frac{\xi_{k-1}-1}{\xi_k}(\boldsymbol{v}_t^{(k)} - \boldsymbol{v}_t^{(k-1)})$;
10     $\widetilde{\boldsymbol{x}}_t^{(k)} = \boldsymbol{x}_t^{(k)} + \frac{\xi_{k-1}-1}{\xi_k}(\boldsymbol{x}_t^{(k)} - \boldsymbol{x}_t^{(k-1)})$;
11     $\nabla_{\boldsymbol{v}_t} f(\widetilde{\boldsymbol{v}}_t^{(k)}, \widetilde{\boldsymbol{x}}_t^{(k)}) = \nabla_{\boldsymbol{x}_t} f(\widetilde{\boldsymbol{v}}_t^{(k)}, \widetilde{\boldsymbol{x}}_t^{(k)}) = \boldsymbol{\Phi}^{\mathrm{T}}\Big(\boldsymbol{\Phi}(\widetilde{\boldsymbol{v}}_t^{(k)} + \widetilde{\boldsymbol{x}}_t^{(k)}) - \boldsymbol{y}_t\Big)$;
12     $(\boldsymbol{U}_t, \boldsymbol{\Sigma}_t, \boldsymbol{V}_t) = \mathrm{incSVD}\Big(\Big[\boldsymbol{B}_{t-1}\ \Big(\widetilde{\boldsymbol{v}}_t^{(k)} - \frac{1}{2}\nabla_{\boldsymbol{v}_t} f(\widetilde{\boldsymbol{v}}_t^{(k)}, \widetilde{\boldsymbol{x}}_t^{(k)})\Big)\Big]\Big)$;
13     $\boldsymbol{\Theta}_t = \boldsymbol{U}_t \boldsymbol{\Gamma}_{\frac{\mu_k}{2} g_1}(\boldsymbol{\Sigma}_t)\boldsymbol{V}_t^T$;
14     $\boldsymbol{v}_t^{(k+1)} = \boldsymbol{\Theta}_t(:, \mathrm{end})$;
15     $\boldsymbol{x}_t^{(k+1)} = \Gamma_{\frac{\mu_k}{2} g}\Big(\widetilde{\boldsymbol{x}}_t^{(k)} - \frac{1}{2}\nabla_{\boldsymbol{x}_t} f(\widetilde{\boldsymbol{v}}_t^{(k)}, \widetilde{\boldsymbol{x}}_t^{(k)})\Big)$; where $\Gamma_{\frac{\mu_k}{2} g}(\cdot)$ is given as in RAMSIA [48];
       // Compute the updated weights [48].
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     $\beta_j = \frac{\Big(||\mathbf{W}_j(\boldsymbol{x}_t^{(k+1)} - \boldsymbol{z}_j)||_1 + \epsilon\Big)^{-1}}{\sum_{l=0}^{J}\Big(||\mathbf{W}_l(\boldsymbol{x}_t^{(k+1)} - \boldsymbol{z}_l)||_1 + \epsilon\Big)^{-1}}$;
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 $\boldsymbol{Z}_t := \{\boldsymbol{z}_j = \boldsymbol{x}_{t-J+j}^{(k+1)}\}_{j=1}^{J}$;
22 $\boldsymbol{B}_t = \boldsymbol{U}_t(:, 1:d)\boldsymbol{\Gamma}_{\frac{\mu_k}{2} g_1}(\boldsymbol{\Sigma}_t)(1:d, 1:d)\boldsymbol{V}_t(:, 1:d)^{\mathrm{T}}$;
23 return $\widehat{\boldsymbol{x}}_t = \boldsymbol{x}_t^{(k+1)}, \widehat{\boldsymbol{v}}_t = \boldsymbol{v}_t^{(k+1)}, \boldsymbol{Z}_t, \boldsymbol{B}_t$;
```

Considering a point or a pixel i in the given frame, the horizontal and vertical components $\mathrm{v}_{1\mathrm{x}i}$ and $\mathrm{v}_{1\mathrm{y}i}$ of corresponding horizontal and vertical flow vectors $\mathbf{v}_{1\mathrm{x}}$ and $\mathbf{v}_{1\mathrm{y}}$ are obtained, as mentioned in [52], by solving:

$$I_{1\mathrm{x}} \cdot \mathrm{v}_{1\mathrm{x}i} + I_{1\mathrm{y}} \cdot \mathrm{v}_{1\mathrm{y}i} + I_{1t} = 0, \quad (7)$$

where $I_{1\mathrm{x}} = \partial I_1/\partial \mathrm{x}$ and $I_{1\mathrm{y}} = \partial I_1/\partial \mathrm{y}$ 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 $\mathrm{v}_{1\mathrm{x}i}$ and $\mathrm{v}_{1\mathrm{y}i}$. 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 $(\mathbf{v}_{1\mathrm{x}}, \mathbf{v}_{1\mathrm{y}})$, i.e., the horizontal and the vertical components of the optical flow vector. Similarly, we obtain $(\mathbf{v}_{2\mathrm{x}}, \mathbf{v}_{2\mathrm{y}})$.

The estimated motions in the form of optical flow vectors are used along with the frame $\boldsymbol{x}_{t-1}$ 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 $(\mathbf{v}_{1\mathrm{x}}, \mathbf{v}_{1\mathrm{y}})$ and $(\frac{1}{2}\mathbf{v}_{2\mathrm{x}}, \frac{1}{2}\mathbf{v}_{2\mathrm{y}})$ to predict motions in the next frame and compensate them on the frame $\boldsymbol{x}_{t-1}$ to obtain $\boldsymbol{x}'_{t-2}$ and $\boldsymbol{x}'_{t-3}$ respectively. It should be noted that $\boldsymbol{x}'_{t-3}$ is obtained by compensating for half the motions, i.e., $(\frac{1}{2}\mathbf{v}_{2\mathrm{x}}, \frac{1}{2}\mathbf{v}_{2\mathrm{y}})$, between $\boldsymbol{x}_{t-1}$ and $\boldsymbol{x}_{t-3}$. These improved frames $\boldsymbol{x}'_{t-2}$, $\boldsymbol{x}'_{t-3}$ are more correlated to the current frame $\boldsymbol{x}_t$ than $\boldsymbol{x}_{t-2}$, $\boldsymbol{x}_{t-3}$, i.e., without motion estimation and compensation. We also keep the most recent frame $\boldsymbol{x}'_{t-1} = \boldsymbol{x}_{t-1}$ (in Line 4) as one of the prior frames.

Thereafter, $\boldsymbol{v}_t^{(k+1)}$ and $\boldsymbol{x}_t^{(k+1)}$ are iteratively computed as in Lines 14 and 15 in Algorithm 1. It can be noted that the proximal operator $\boldsymbol{\Gamma}_{\tau g_1}(\cdot)$ in Line 13 of Algorithm 1 is defined [7] as:

$$\boldsymbol{\Gamma}_{\tau g_1}(\boldsymbol{X}) = \arg\min_{\boldsymbol{V}} \left\{ \tau g_1(\boldsymbol{V}) + \frac{1}{2}||\boldsymbol{V} - \boldsymbol{X}||_2^2 \right\}, \tag{8}$$

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 and 17). As suggested in [2], the convergence of Algorithm 1 in Line 8 is determined by evaluating the criterion $\|\partial H(\boldsymbol{x}_t, \boldsymbol{v}_t)|_{\boldsymbol{x}_t^{(k+1)}, \boldsymbol{v}_t^{(k+1)}}\|_2^2 < 2 \times 10^{-7} \|(\boldsymbol{x}_t^{(k+1)}, \boldsymbol{v}_t^{(k+1)})\|_2^2$. In the next step, we perform an update of the priors $\boldsymbol{Z}_t$ and $\boldsymbol{B}_t$.

2.2.3. Prior Update

The update of $\boldsymbol{Z}_t$ and $\boldsymbol{B}_t$ [7] is carried out after each time instance (see Lines 21 and 22, Algorithm 1). Due to the correlation between subsequent frames, we update the prior information $\boldsymbol{Z}_t$ by using the J latest recovered sparse components, which is given by, $\boldsymbol{Z}_t := \{\boldsymbol{z}_j = \boldsymbol{x}_{t-J+j}\}_{j=1}^{J}$. 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, the incremental singular value decomposition [47] method (incSVD($\cdot$) in Line 12, Algorithm 1) is used. It is worth noting that the update $\boldsymbol{B}_t = \boldsymbol{U}_t \boldsymbol{\Gamma}_{\frac{\mu_k}{2} g_1}(\boldsymbol{\Sigma}_t)\boldsymbol{V}_t^{\mathrm{T}}$, causes the dimension of $\boldsymbol{B}_t$ to increase as $\boldsymbol{B}_t \in \mathbb{R}^{n\times(d+1)}$ after each instance. However, in order to maintain a reasonable number of d, we take $\boldsymbol{B}_t = \boldsymbol{U}_t(:, 1:d)\boldsymbol{\Gamma}_{\frac{\mu_k}{2} g_1}(\boldsymbol{\Sigma}_t)(1:d, 1:d)\boldsymbol{V}_t(:, 1:d)^{\mathrm{T}}$. The computational cost of incSVD($\cdot$) is lower than conventional SVD [8,47] 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$[\boldsymbol{B}_{t-1}\ \boldsymbol{v}_t]$, 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 $\boldsymbol{B}_{t-1} \in \mathbb{R}^{n\times d}$ to obtain $\boldsymbol{B}_{t-1} = \boldsymbol{U}_{t-1}\boldsymbol{\Sigma}_{t-1}\boldsymbol{V}_{t-1}^{\mathrm{T}}$. Therefore, we can derive $(\boldsymbol{U}_t, \boldsymbol{\Sigma}_t, \boldsymbol{V}_t)$ via $(\boldsymbol{U}_{t-1}, \boldsymbol{\Sigma}_{t-1}, \boldsymbol{V}_{t-1})$ and $\boldsymbol{v}_t$. We write the matrix $[\boldsymbol{B}_{t-1}\ \boldsymbol{v}_t]$ as

$$[\boldsymbol{B}_{t-1}\ \boldsymbol{v}_t] = \left[\boldsymbol{U}_{t-1}\ \ \frac{\boldsymbol{\delta}_t}{\|\boldsymbol{\delta}_t\|_2}\right] \cdot \begin{bmatrix} \boldsymbol{\Sigma}_{t-1} & \boldsymbol{e}_t \\ \mathbf{0}^{\mathrm{T}} & \|\boldsymbol{\delta}_t\|_2 \end{bmatrix} \cdot \begin{bmatrix} \boldsymbol{V}_{t-1}^{\mathrm{T}} & \mathbf{0} \\ \mathbf{0}^{\mathrm{T}} & 1 \end{bmatrix}, \tag{9}$$

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 computing the SVD of the central term of (9), we obtain $\begin{bmatrix} \boldsymbol{\Sigma}_{t-1} & \boldsymbol{e}_t \\ \boldsymbol{0}^{\mathrm{T}} & \|\boldsymbol{\delta}_t\|_2 \end{bmatrix} = \widetilde{\boldsymbol{U}}\widetilde{\boldsymbol{\Sigma}}\widetilde{\boldsymbol{V}}^{\mathrm{T}}$. Eventually, we obtain $\boldsymbol{U}_t = \left[\boldsymbol{U}_{t-1} \ \frac{\boldsymbol{\delta}_t}{\|\boldsymbol{\delta}_t\|_2}\right] \cdot \widetilde{\boldsymbol{U}}$, $\boldsymbol{\Sigma}_t = \widetilde{\boldsymbol{\Sigma}}$, and also, $\boldsymbol{V}_t = \begin{bmatrix} \boldsymbol{V}_{t-1}^{\mathrm{T}} & \boldsymbol{0} \\ \boldsymbol{0}^{\mathrm{T}} & 1 \end{bmatrix} \cdot \widetilde{\boldsymbol{V}}$.

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 (Linux) with Intel i5 3.5 GHz CPU (4 cores) and 12GB RAM. The C++ implementation was carried out and tested on a computer (Windows) with i7-4510U 2.0 GHz CPU (2 cores) and 8 GB memory. By doing that, the cross platform functionality can also be verified with these two machines.

For the experimental evaluation with the various existing methods, mainly two sequences [53], `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 Section 2.2.3.

We evaluate the performance of the proposed CORPCA-OF in Algorithm 1 and compare it against the existing methods, Robust Principal Component Analysis (RPCA) [1], Grassmannian Robust Adaptive Subspace Tracking Algorithm(GRASTA) [11], and Recursive Projected Compressive Sensing(ReProCS) [17,22]. RPCA [1] is a batch-based method assuming full access to the data, while GRASTA [11] and ReProCS [17] 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 [7] 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. Figure 3 shows a few examples of the prior information generated for the sequences `Bootstrap` and `Curtain`. In Figure 3a, 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 Figure 3b–d, . Especially in Figure 3c, the generated frames #448′ and #449′ have significantly improved due to dense motion compensations. In Figure 3d, 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 [7], RPCA [1], GRASTA [11], and ReProCS [17]. 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 [7] as well as for GRASTA [11] and ReProCS [17]. Three latest previous foregrounds are used as the foreground prior for CORPCA. Meanwhile, CORPCA-OF refines them using optical flow [6].

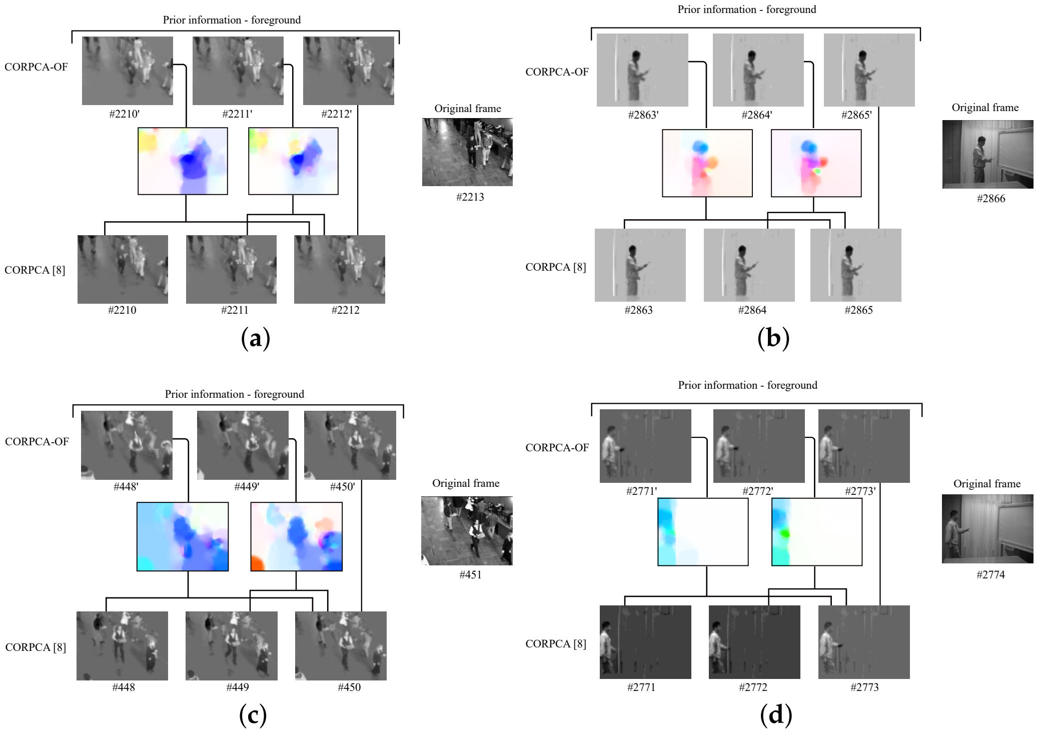


Figure 3. Prior information generation in CORPCA-OF using optical flow [6]. (**a**) `Bootstrap` #2213; (**b**) `Curtain` #2866; (**c**) `Bootstrap` #451; (**d**) `Curtain` #2774.

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 Figure 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.

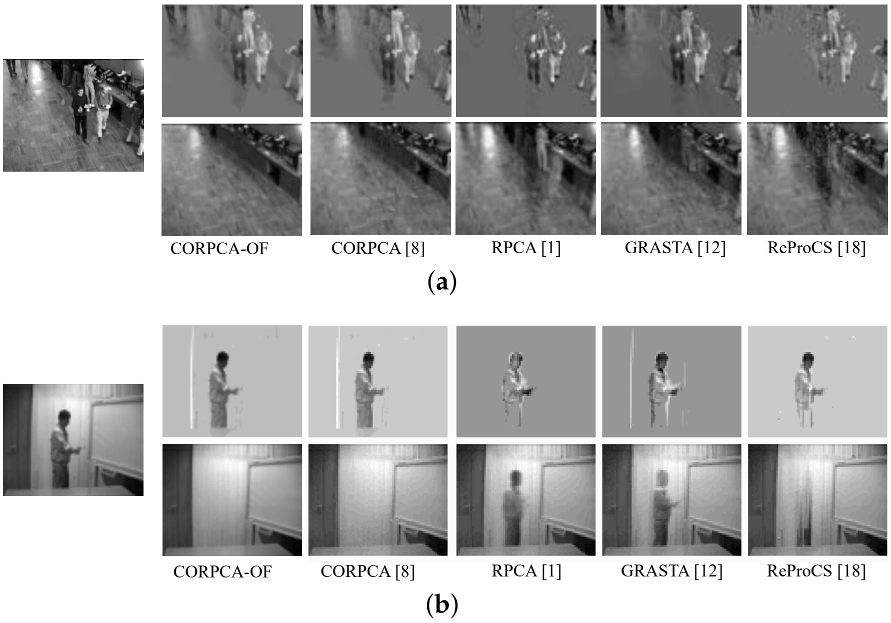


Figure 4. Foreground and background separation for the different separation methods with full data access `Bootstrap` #2213 and `Curtain` #2866. (**a**) `Bootstrap`; (**b**) `Curtain`.

Additionally, we also compare the visual results of CORPCA-OF, CORPCA and ReProCS for the frames `Bootstrap` #2213 (in Figure 5) and for `Curtain` #2866 (in Figure 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 Figures 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 Figure 5a) or $m/n = 0.4$ (in Figure 6a), the extracted foreground frames of CORPCA-OF are better than those of 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 Figures 5 and 6 that the visual results obtained with CORPCA-OF are of superior quality compared to ReProCS and have significant improvements over CORPCA.

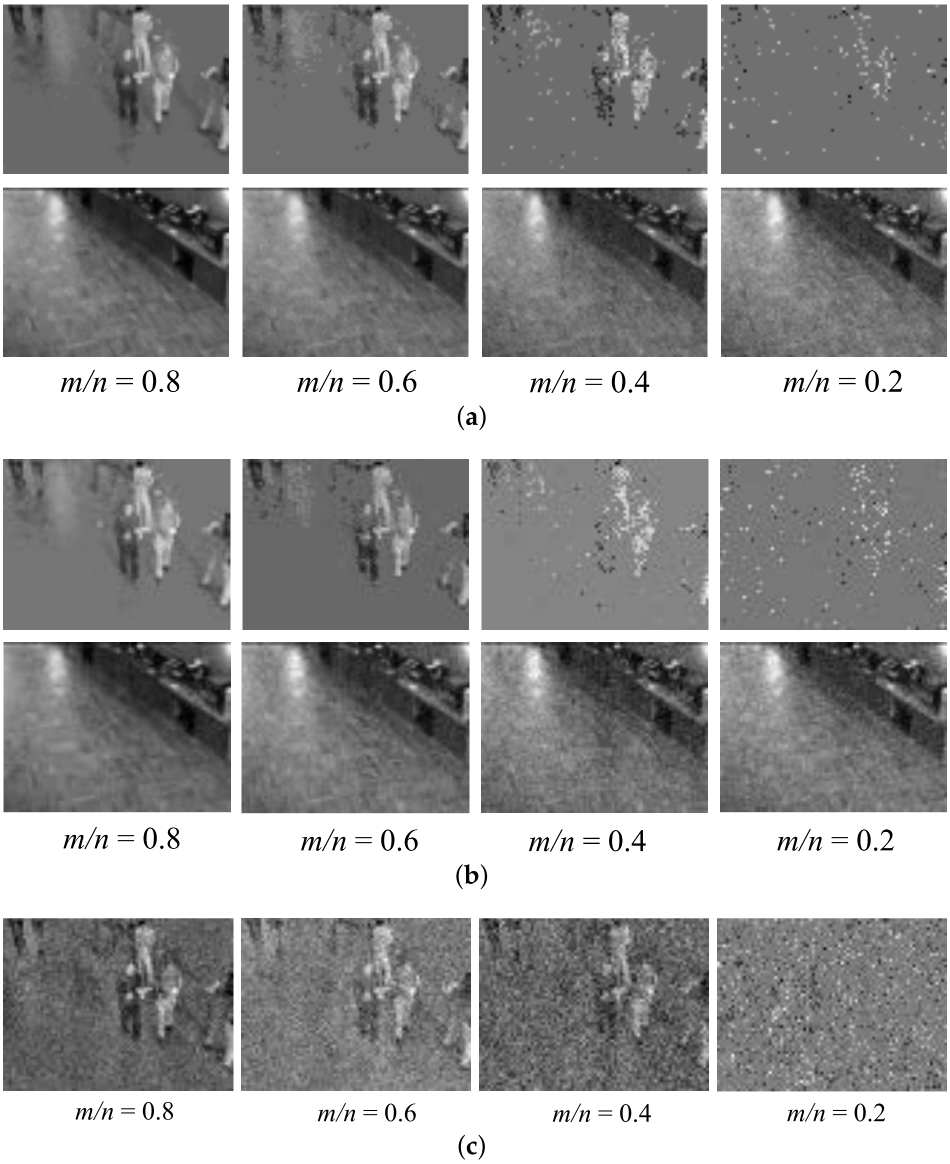
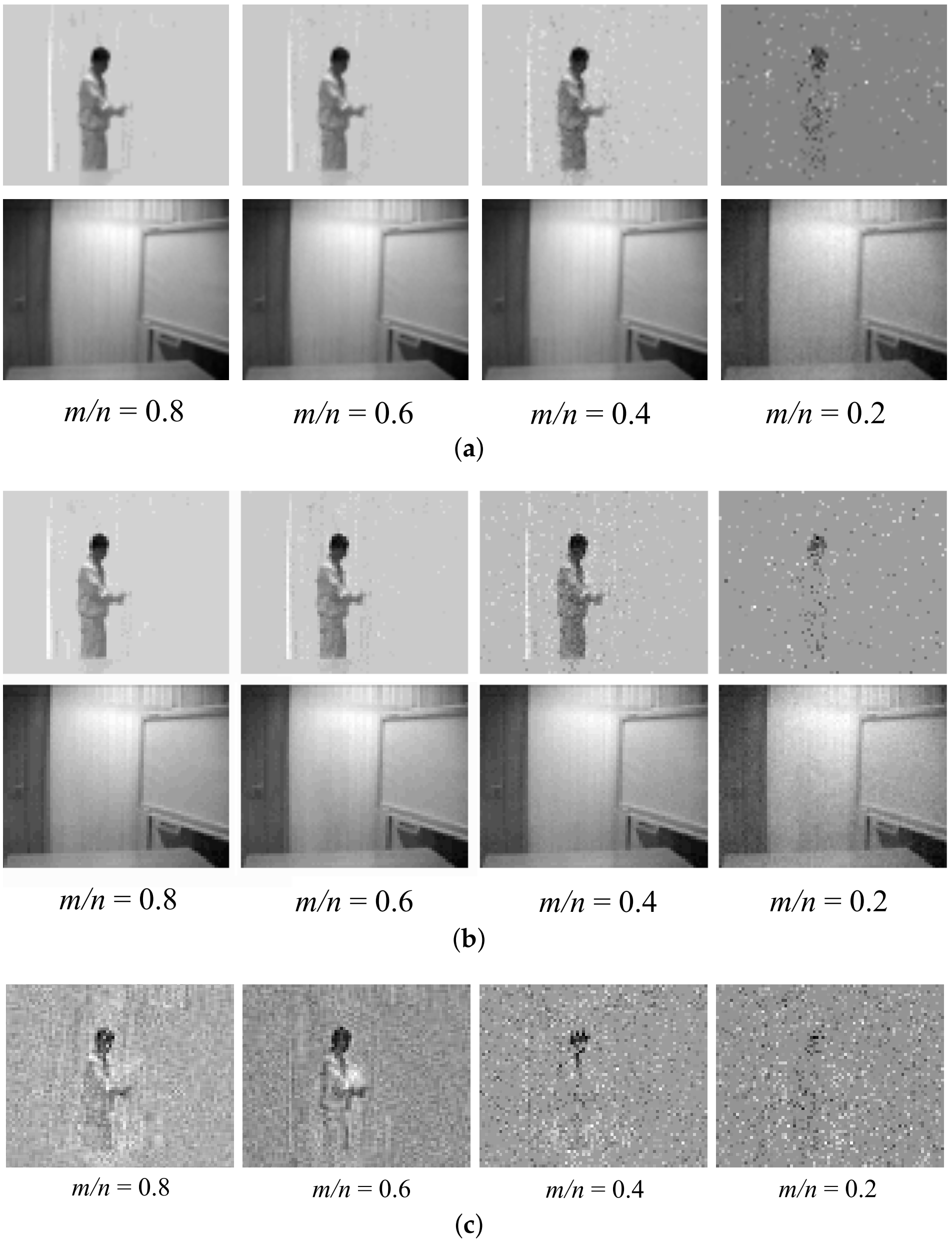


Figure 5. Compressive foreground and background separation of CORPCA-OF, CORPCA [7], and ReProCS [17] with different measurement rates m/n of frame `Bootstrap` #2213. (**a**) CORPCA-OF; (**b**) CORPCA [7]; (**c**) ReProCS [17].



$m/n = 0.8$ $m/n = 0.6$ $m/n = 0.4$ $m/n = 0.2$

(**a**)

$m/n = 0.8$ $m/n = 0.6$ $m/n = 0.4$ $m/n = 0.2$

(**b**)

$m/n = 0.8$ $m/n = 0.6$ $m/n = 0.4$ $m/n = 0.2$

(**c**)

Figure 6. Compressive foreground and background separation of CORPCA-OF, CORPCA [7], and ReProCS [17] with different measurement rates m/n of frame `Curtain` #2866. (**a**) CORPCA-OF; (**b**) CORPCA [7]; (**c**) ReProCS [17].

3.2.2. Quantitative Results

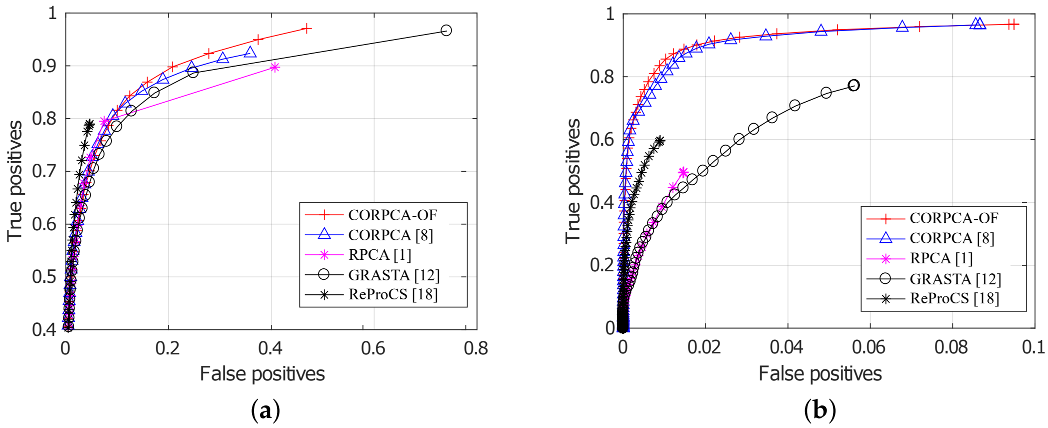
We evaluate quantitatively the separation performance via the *receiver operating curve* (ROC) metric [54]. The metrics *True positives* and *False positives* are defined in [54] as:

$$True\ positives = \frac{|\{\text{Foreground}\} \cap \{\text{Groundtruth Foreground}\}|}{|\{\text{Groundtruth Foreground}\}|} \tag{10}$$

$$False\ positives = \frac{|\{\text{Foreground}\} \cap \{\text{Groundtruth Background}\}|}{|\{\text{Groundtruth Background}\}|} \tag{11}$$

For plotting the ROC curve, a set of foreground frames corresponding to the ground truth frames are selected. Then we threshold the foreground frames at various levels and compute true positives and false positives by comparing with the ground truth frames for each threshold level. We then plot true positives against false positives.

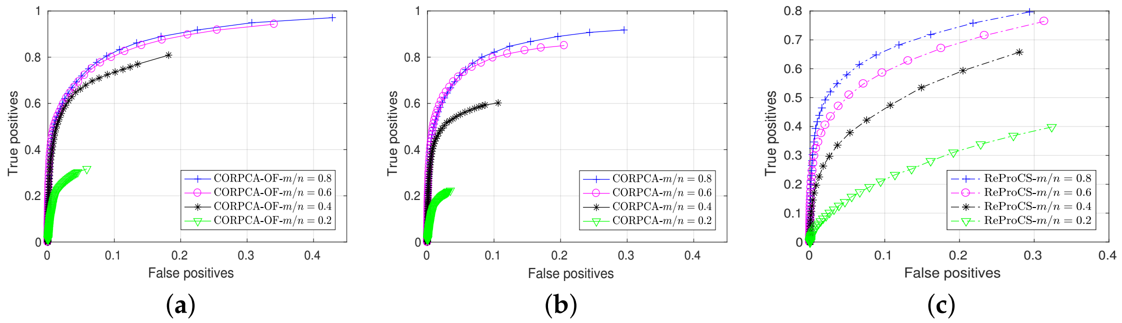
Figure 7 illustrates the ROC results when assuming full data access, i.e., $m/n = 1$, of CORPCA-OF, CORPCA, RPCA, GRASTA, and ReProCS. The results show that CORPCA-OF delivers higher performance than the other methods.



(**a**) (**b**)

Figure 7. ROC for the different separation methods with full data. (**a**) `Bootstrap`; (**b**) `Curtain`.

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



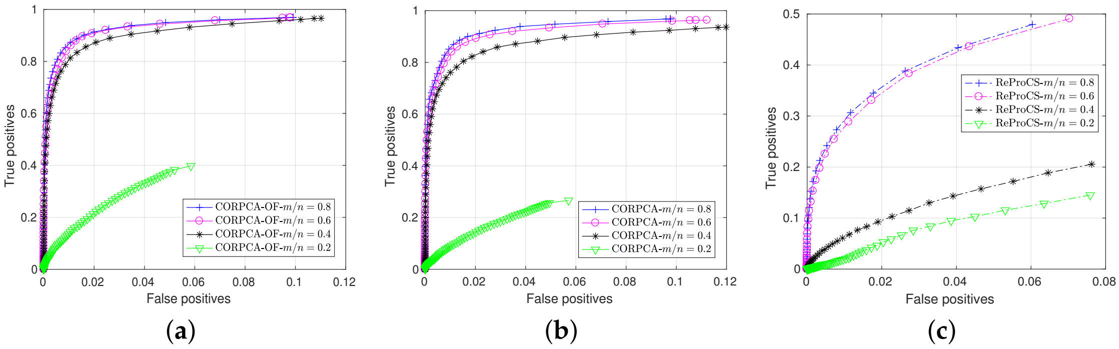
(**a**) (**b**) (**c**)

Figure 8. ROC for CORPCA-OF, CORPCA [7], and ReProCS [17] with compressive measurement rates m/n for `Bootstrap` sequence. (**a**) CORPCA-OF; (**b**) CORPCA [7]; (**c**) ReProCS [17].

3.3. Additional Results

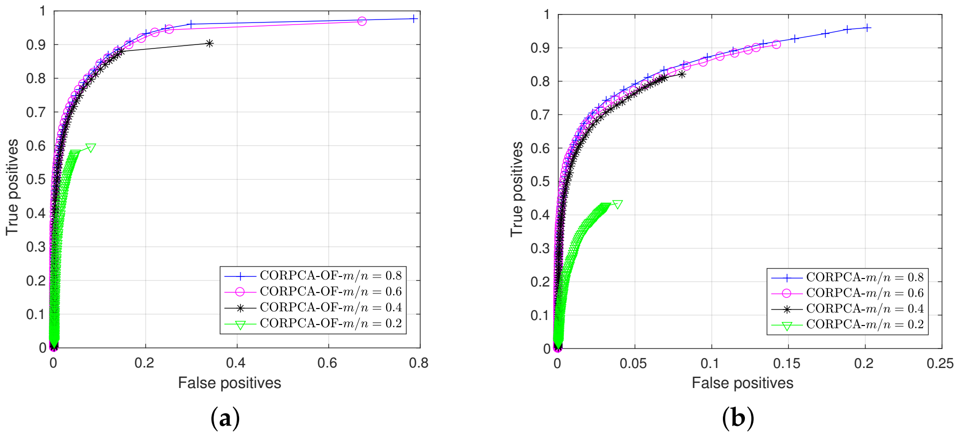
3.3.1. `Escalator` and `Fountain` sequences

The CORPCA-OF method was also compared with CORPCA against `Escalator` and `Fountain` sequences for compressive measurements. From Figure 10a,b, it is clear that CORPCA-OF performs slightly better than CORPCA. In Figure 11a,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.



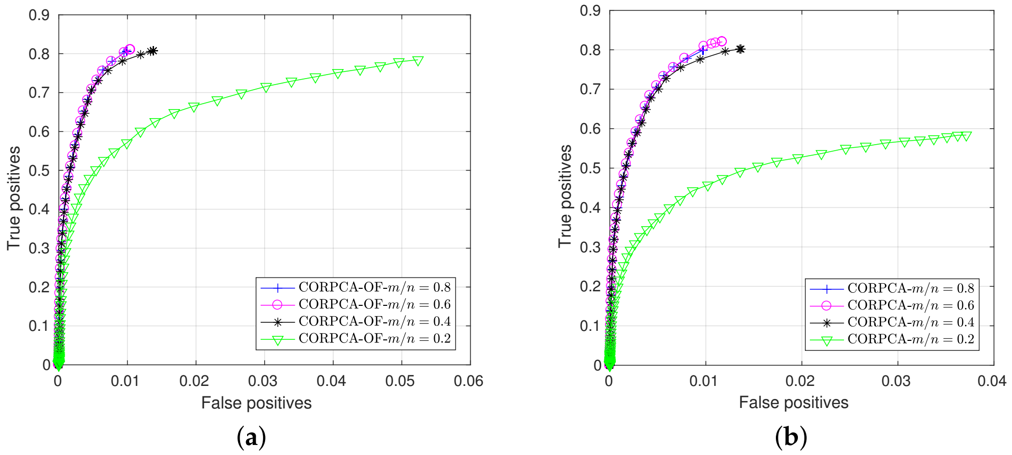
(**a**) (**b**) (**c**)

Figure 9. ROC for CORPCA-OF, CORPCA [7], and ReProCS [17] with compressive measurement rates m/n for `Curtain` sequence. (**a**) CORPCA-OF; (**b**) CORPCA [7]; (**c**) ReProCS [17].



(**a**) (**b**)

Figure 10. ROC of CORPCA-OF and CORPCA [7] methods with different rates for `Escalator` sequence. (**a**) CORPCA-OF; (**b**) CORPCA [7].



(**a**) (**b**)

Figure 11. ROC of CORPCA-OF and CORPCA [7] methods with different rates for `Fountain` sequence. (**a**) CORPCA-OF; (**b**) CORPCA [7].

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 Figure12. In Figure13, 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 Figure 12a over CORPCA in Figure 12b. But in case of Curtain sequence, at low rates $m/n = \{0.4; 0.2\}$, there is significant improvement of foreground in Figure 13a for CORPCA-OF over CORPCA in Figure 13b.

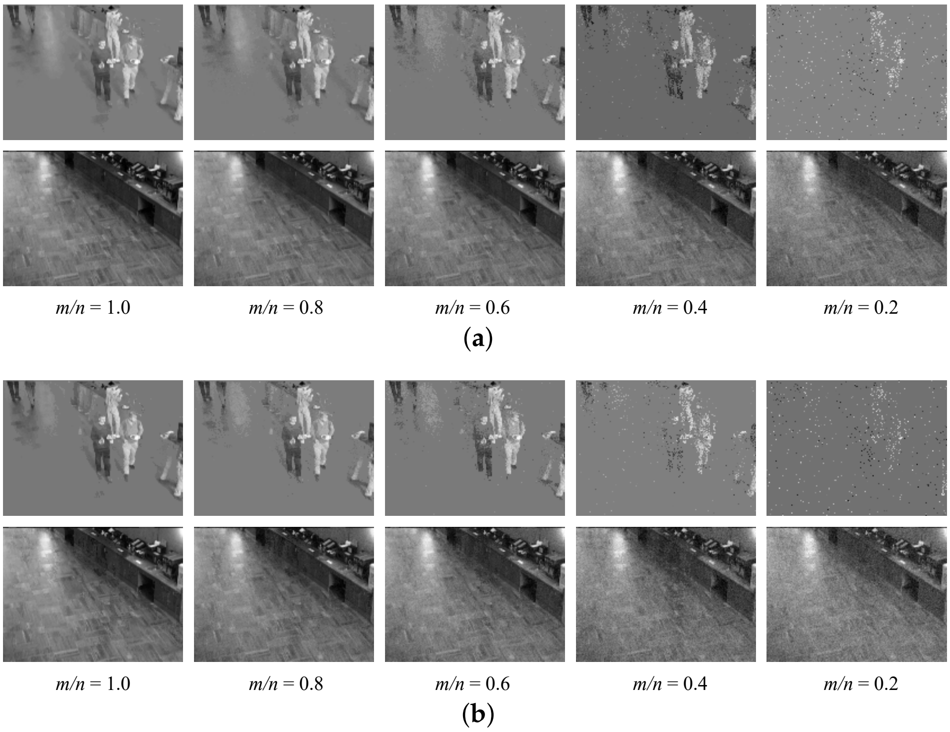
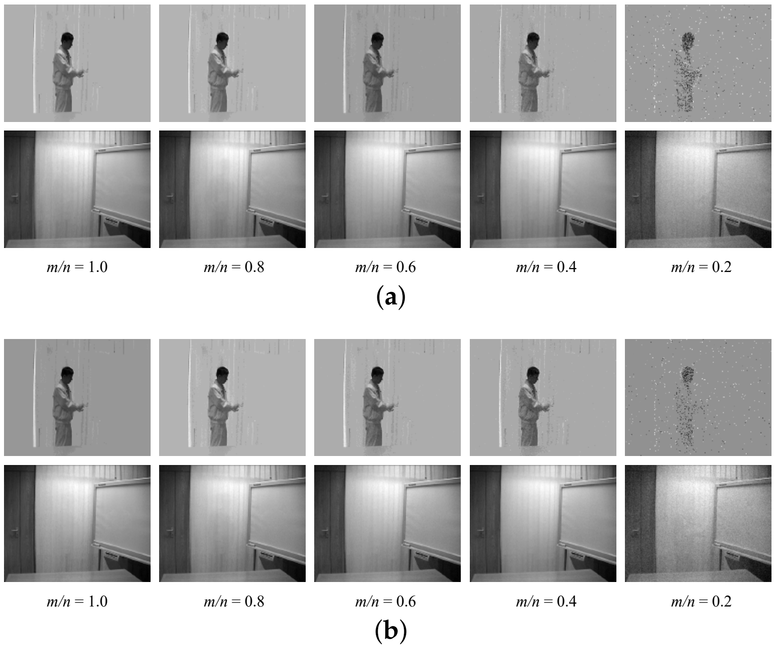


Figure 12. Comparison of CORPCA-OF and CORPCA [7] at full resolution with different rates m/n for Bootstrap #2213. (**a**) CORPCA-OF; (**b**) CORPCA [7].

3.3.3. Separation Results with Various Datasets

CDnet dataset: We tested our algorithm with some sequences from the CDnet dataset [39]. tramCrossroad1 is a sequence captured at low frame rate. The foreground and backgrouns separation result for frame #655 is shown in Figure 14a. A thermal imaging sequence corridor was seperated using CORPCA-OF and the result for frame #686 is shown in Figure 14b. The separation results for a sequence with camera jitter, badminton is shown in Figure 14c. canoe is a challenging sequence with background motion or in other words, with dynamic background. The results for this sequence is shown in Figure 14d for frame #145. We can observe some artifacts of the canoe from foreground in the background image. This is because of the complex motion of water. The sequences badminton and canoe are also part of the SBMNet dataset.

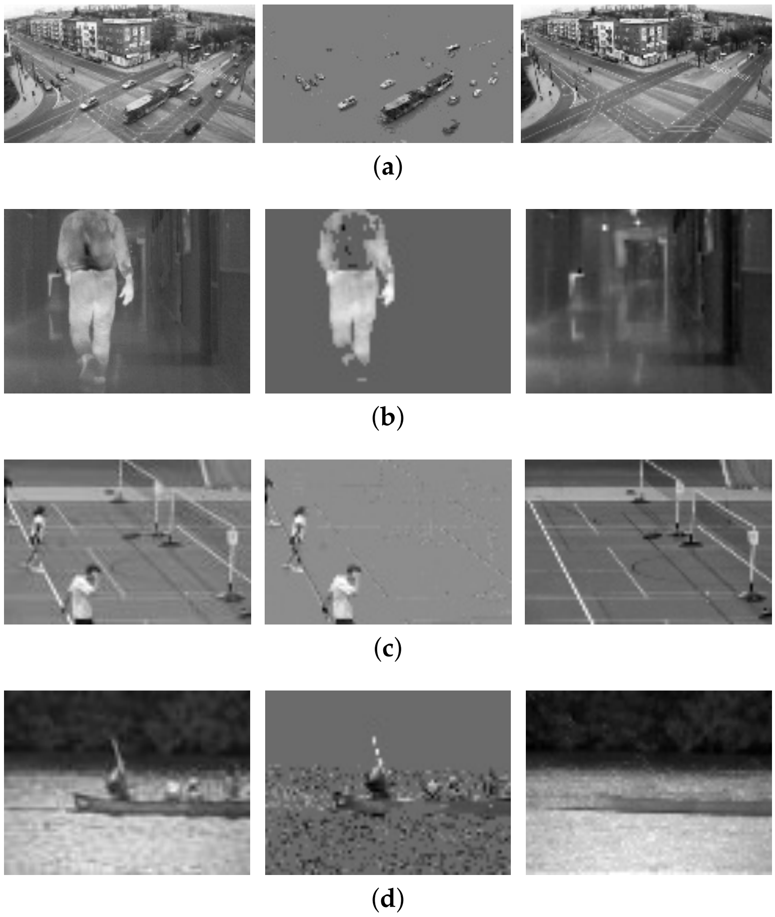
SBMNet dataset: The results for sequences from SBMNet dataset [42] are shown in Figure 15. MPEG4_40 is an animated sequence of cars at a traffic signal. It can be seen from Figure 15a that our algorithm works for synthetic images as well. A cluttered sequence IndianTraffic3 was tested and the result for frame #622 can be seen from Figure 15b. We also tested our algorithm on an underwater sequence Hybrid and the result is shown in Figure 15c.



(**a**)

(**b**)

Figure 13. Comparison of CORPCA-OF and CORPCA [7] at full resolution with different rates m/n for `Curtain` #2866. (**a**) CORPCA-OF; (**b**) CORPCA [7].



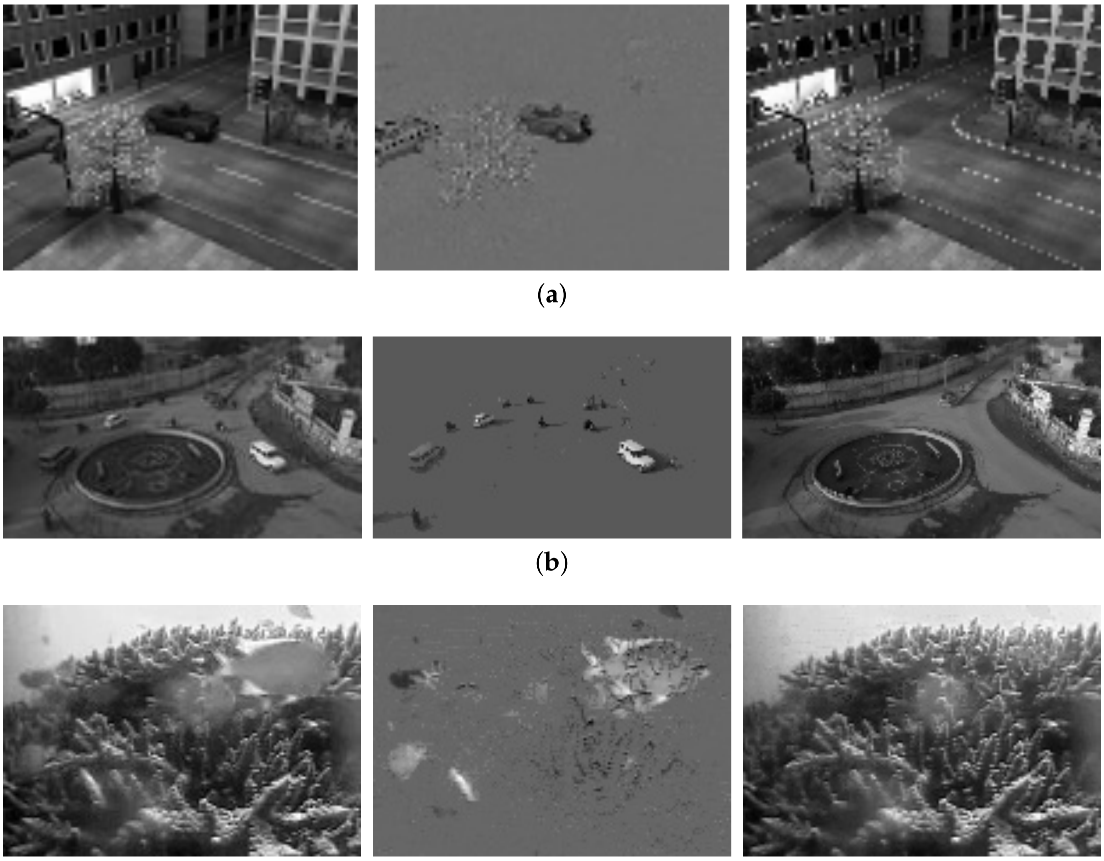
(**a**)

(**b**)

(**c**)

(**d**)

Figure 14. Foreground and background separation results of sequences from CDnet dataset [39]. (**a**) `tramCrossroad1_fps #655`; (**b**) `corridor #686`; (**c**) `badminton #471`; (**d**) `canoe #145`.



(**a**)

(**b**)

(**c**)

Figure 15. Foreground and background separation results of sequences from SBMNet dataset [42]. (**a**) `MPEG4_40 #193`; (**b**) `IndianTraffic3 #622`; (**c**) `Hybrid #4`.

These results demonstrate the versatility of our algorithm.

`HighwayI`**:** This sequence is from the ATON dataset [55]. 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 Figure 16.

`Laboratory`**:** This sequence is from the ATON dataset [55]. It contains an indoor sequence with two persons appearing for certain intervals. Original frames #142 to #158 are shown in Figure17 along with the separated background and foreground frames.

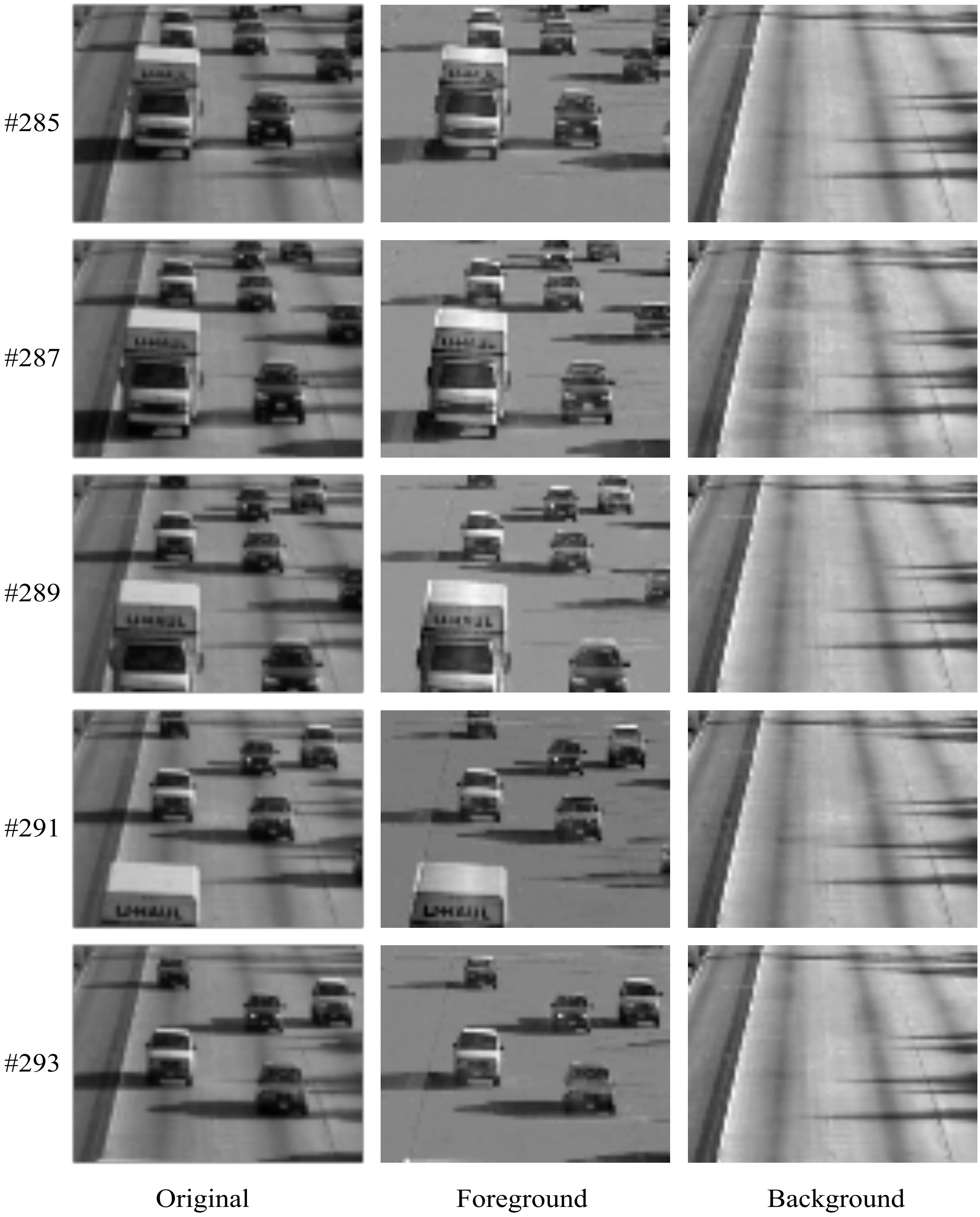


Figure 16. Foreground and background separation for `HighwayI` sequence.

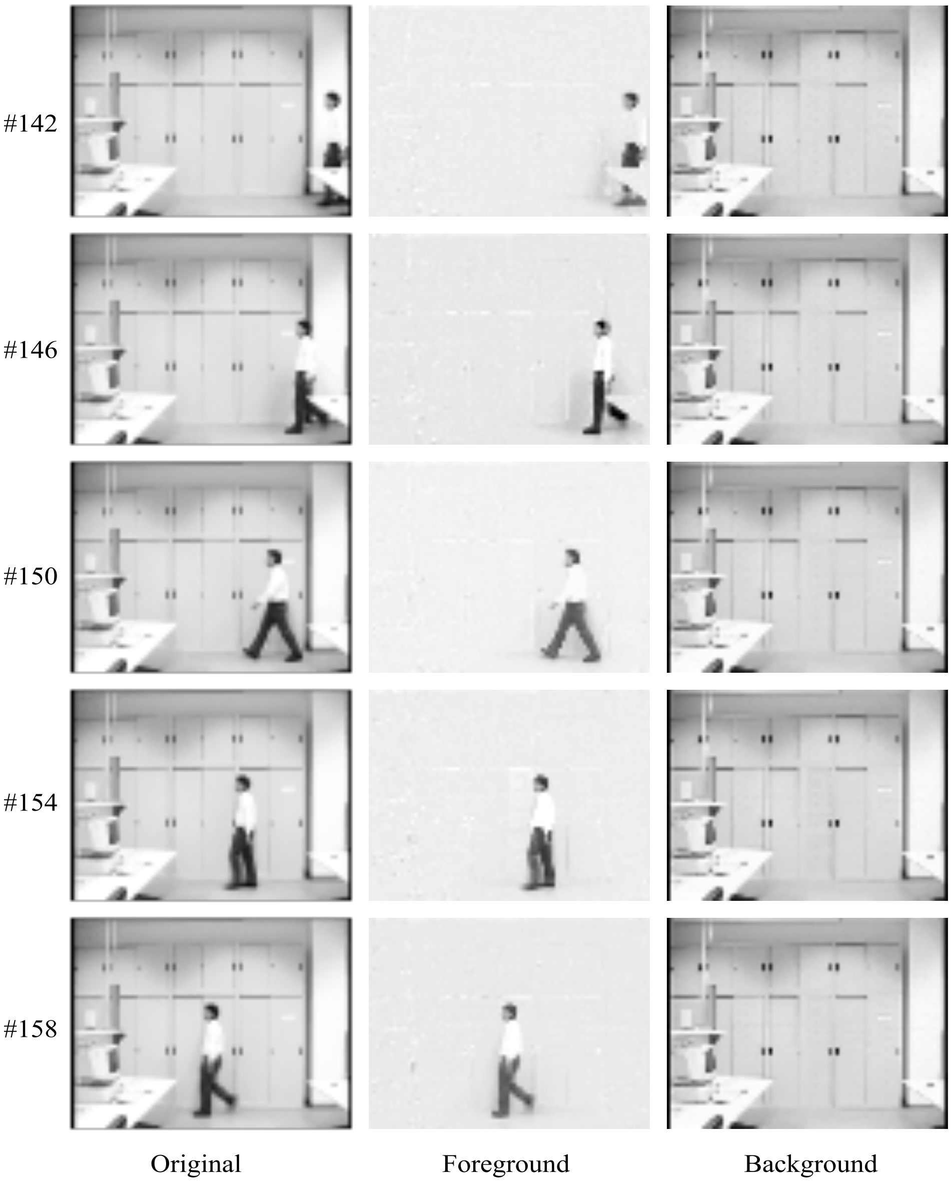


Figure 17. Foreground and background separation for `Laboratory` sequence.

4. Conclusions

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-ℓ_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 (see Appendix A) of CORPCA-OF

(The code for the CORPCA-OF algorithm is available at [56]) has been carried out, where the methods can be used as libraries for video separation. For separating a frame of resolution 80 × 60 and full data ($m/n = 1$), on an average, the Matlab implementation takes about 8 s per frame, out of which about 0.4 s is for the computation of optical flow. The same algorithm on C++ takes about 28 s per frame, with about 0.15 s for the optical flow computation. Building on this, the next step would be to make the algorithm more robust, adaptive, dynamic, and real-time by using Graphics Processing Units (GPUs) so that the algorithm can handle larger images.

Author Contributions: The work was carried out by S.P. based on the research and guidance of H.V.L. in collaboration with T.H.L. and the supervision of A.K.

Funding: This research received no external funding.

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.

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

Abbreviations

The following abbreviations are used in this paper:

CDnet	ChangeDetection.NET
CORPCA	Compressive Online Robust Principal Component Analysis
CORPCA-OF	Compressive Online Robust Principal Component Analysis with Optical Flow
GPU	Graphics 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
SBMnet	SceneBackgroundModeling.NET
SVD	Singular Value Decomposition

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

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

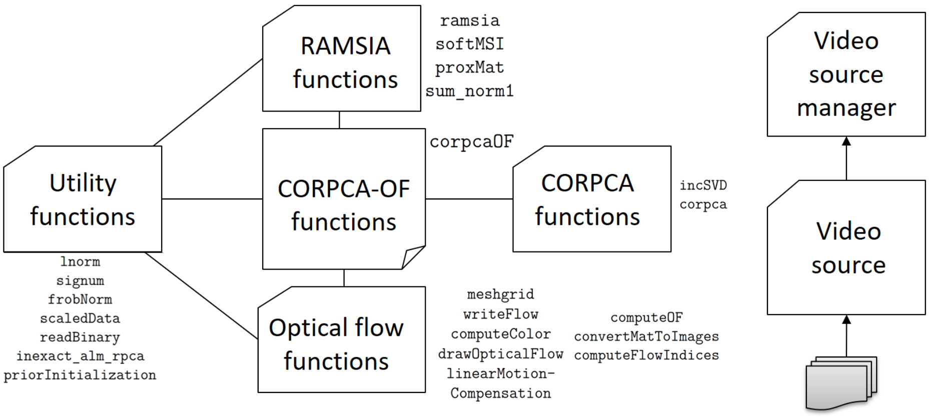
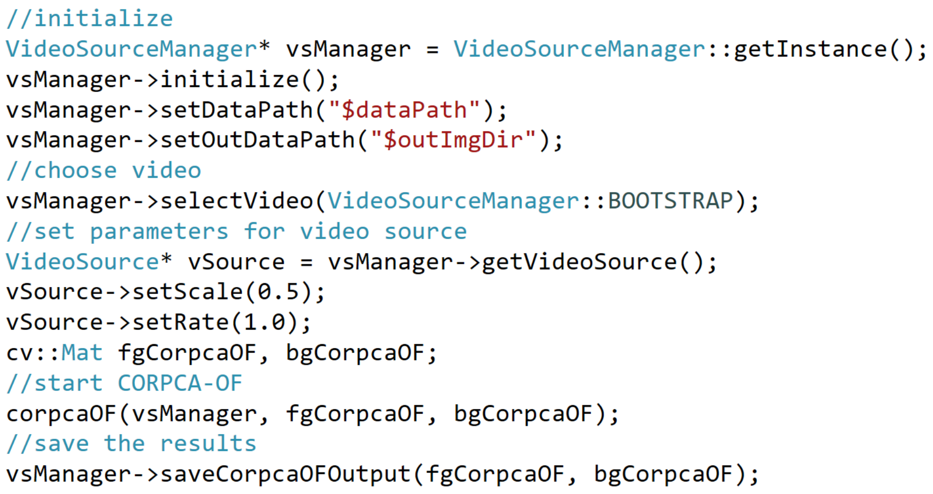


Figure A1. Overview of CORPCA-OF implementation.



```
//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? *JACM* **2011**, *58*, 11. [CrossRef]
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. *Adv. Neural Inf. Process. Syst.* **2009**, 2080–2088.
3. Horn, B.K.; Schunck, B.G. Determining optical flow. *Artif. Intell.* **1981**, *17*, 185–203. [CrossRef]
4. Bruhn, A.; Weickert, J.; Schnörr, C. Lucas/Kanade meets Horn/Schunck: Combining local and global optic flow methods. *Int. J. Comput. Vis.* **2005**, *61*, 211–231. [CrossRef]
5. Baker, S.; Scharstein, D.; Lewis, J.; Roth, S.; Black, M.J.; Szeliski, R. A database and evaluation methodology for optical flow. *Int. J. Comput. Vis.* **2011**, *92*, 1–31. [CrossRef]
6. Brox, T.; Malik, J. Large displacement optical flow: Descriptor matching in variational motion estimation. *IEEE Trans. Pattern Anal. Mach. Intell.* **2011**, *33*, 500–513. [CrossRef] [PubMed]
7. Luong, H.V.; Deligiannis, N.; Seiler, J.; Forchhammer, S.; Kaup, A. Compressive online robust principal component analysis with multiple prior information. In Proceedings of the IEEE Global Conference on Signal and Information Processing, Montreal, QC, Canada, 14–16 November 2017.
8. Rodriguez, P.; Wohlberg, B. Incremental principal component pursuit for video background modeling. *J. Math. Imaging Vis.* **2016**, *55*, 1–18. [CrossRef]
9. 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. [CrossRef]
10. Xiao, W.; Huang, X.; Silva, J.; Emrani, S.; Chaudhuri, A. Online Robust Principal Component Analysis with Change Point Detection. *arXiv* **2017**, arxiv:1702.05698.
11. He, J.; Balzano, L.; Szlam, A. Incremental gradient on the grassmannian for online foreground and background separation in subsampled video. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Providence, RI, USA, 16–21 June 2012; pp. 1568–1575.
12. Xu, J.; Ithapu, V.K.; Mukherjee, L.; Rehg, J.M.; Singh, V. Gosus: Grassmannian online subspace updates with structured-sparsity. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), Sydney, Australia, 1–8 December 2013; pp. 3376–3383.
13. Feng, J.; Xu, H.; Yan, S. Online robust pca via stochastic optimization. *Adv. Neural Inf. Process. Syst.* **2013**, *26*, 404–412.
14. Mansour, H.; Jiang, X. A robust online subspace estimation and tracking algorithm. In Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Brisbane, Australia, 19–24 April 2015; pp. 4065–4069.
15. Wright, J.; Ganesh, A.; Min, K.; Ma, Y. Compressive principal component pursuit. *Inf. Inference J. IMA* **2013**, *2*, 32–68. [CrossRef]
16. Pan, P.; Feng, J.; Chen, L.; Yang, Y. Online compressed robust PCA. In Proceedings of the International Joint Conference on Neural Networks (IJCNN), Anchorage, AK, USA, 14–19 May 2017; pp. 1041–1048.

17. Guo, H.; Qiu, C.; Vaswani, N. An online algorithm for separating sparse and low-dimensional signal sequences from their sum. *IEEE Trans. Signal Process.* **2014**, *62*, 4284–4297. [CrossRef]
18. 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 Trans. Signal Process.* **2016**, *64*, 3651–3666. [CrossRef]
19. Warnell, G.; Bhattacharya, S.; Chellappa, R.; Basar, T. Adaptive-Rate Compressive Sensing Using Side Information. *IEEE Trans. Image Process.* **2015**, *24*, 3846–3857. [CrossRef] [PubMed]
20. Qiu, C.; Vaswani, N.; Lois, B.; Hogben, L. Recursive Robust PCA or Recursive Sparse Recovery in Large but Structured Noise. *IEEE Trans. Inf. Theory* **2014**, *60*, 5007–5039. [CrossRef]
21. Zhan, J.; Vaswani, N. Robust PCA with partial subspace knowledge. In Proceedings of the 2014 IEEE International Symposium on Information Theory, Honolulu, HI, USA, 29 June–4 July 2014; doi:10.1109/ISIT.2014.6875222.
22. Vaswani, N.; Zhan, J. Recursive Recovery of Sparse Signal Sequences From Compressive Measurements: A Review. *IEEE Trans. Signal Process.* **2016**, *64*, 3523–3549. [CrossRef]
23. 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. [CrossRef]
24. Vaswani, N.; Lu, W. Modified-CS: Modifying compressive sensing for problems with partially known support. *IEEE Trans. Signal Process.* **2010**, *58*, 4595–4607. [CrossRef]
25. Gonzalez, C.G.; Absil, O.; Absil, P.A.; Van Droogenbroeck, M.; Mawet, D.; Surdej, J. Low-rank plus sparse decomposition for exoplanet detection in direct-imaging ADI sequences-The LLSG algorithm. *Astron. Astrophys.* **2016**, *589*, A54. [CrossRef]
26. Sobral, A.; Bouwmans, T.; ZahZah, E.H. Double-constrained RPCA based on saliency maps for foreground detection in automated maritime surveillance. In Proceedings of the 12th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS), Karlsruhe, Germany, 25–28 August 2015; pp. 1–6.
27. Javed, S.; Mahmood, A.; Bouwmans, T.; Jung, S.K. Background–Foreground Modeling Based on Spatiotemporal Sparse Subspace Clustering. *IEEE Trans. Image Process.* **2017**, *26*, 5840–5854. [CrossRef] [PubMed]
28. Javed, S.; Mahmood, A.; Bouwmans, T.; Jung, S.K. Spatiotemporal low-rank modeling for complex scene background initialization. *IEEE Trans. Circuits Syst. Video Technol.* **2016**, *28*, 1315–1329. [CrossRef]
29. Berjón, D.; Cuevas, C.; Morán, F.; García, N. Real-time nonparametric background subtraction with tracking-based foreground update. *Pattern Recognit.* **2018**, *74*, 156–170. [CrossRef]
30. Benezeth, Y.; Jodoin, P.M.; Emile, B.; Laurent, H.; Rosenberger, C. Comparative study of background subtraction algorithms. *J. Electron. Imaging* **2010**, *19*, 033003.
31. Bouwmans, T. Traditional and recent approaches in background modeling for foreground detection: An overview. *Comput. Sci. Rev.* **2014**, *11*, 31–66. [CrossRef]
32. Cuevas, C.; Martínez, R.; García, N. Detection of stationary foreground objects: A survey. *Comput. Vis. Image Underst.* **2016**, *152*, 41–57. [CrossRef]
33. Vacavant, A.; Chateau, T.; Wilhelm, A.; Lequièvre, L. A benchmark dataset for outdoor foreground/background extraction. *Asian Conf. Comput. Vis.* **2012**, *7728*, 291–300.
34. Sobral, A.; Vacavant, A. A comprehensive review of background subtraction algorithms evaluated with synthetic and real videos. *Comput. Vis. Image Underst.* **2014**, *122*, 4–21. [CrossRef]
35. Bianco, S.; Ciocca, G.; Schettini, R. Combination of video change detection algorithms by genetic programming. *IEEE Trans. Evol. Comput.* **2017**, *21*, 914–928. [CrossRef]
36. Braham, M.; Piérard, S.; Van Droogenbroeck, M. Semantic background subtraction. In Proceedings of the IEEE International Conference on Image Processing (ICIP), Beijing, China, 17–20 September 2017; pp. 4552–4556.
37. Lee, S.H.; Kwon, S.C.; Shim, J.W.; Lim, J.E.; Yoo, J. WisenetMD: Motion Detection Using Dynamic Background Region Analysis. *arXiv* **2018**, arxiv:1805.09277.
38. St-Charles, P.L.; Bilodeau, G.A.; Bergevin, R. Universal background subtraction using word consensus models. *IEEE Trans. Image Process.* **2016**, *25*, 4768–4781. [CrossRef]
39. Wang, Y.; Jodoin, P.M.; Porikli, F.; Konrad, J.; Benezeth, Y.; Ishwar, P. CDnet 2014: An expanded change detection benchmark dataset. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Columbus, OH, USA, 23–28 June 2014; pp. 393–400.

40. Bouwmans, T.; Maddalena, L.; Petrosino, A. Scene background initialization: A taxonomy. *Pattern Recognit. Lett.* **2017** , *96*, 3–11. [CrossRef]
41. Jodoin, P.M.; Maddalena, L.; Petrosino, A.; Wang, Y. Extensive benchmark and survey of modeling methods for scene background initialization. *IEEE Trans. Image Process.* **2017**, *26*, 5244–5256. [CrossRef] [PubMed]
42. Maddalena, L.; Bouwmans, T. Scene Background Modeling and Initialization (SBMI) Workshop. Genova, Italy. 2015. Available online: http://sbmi2015.na.icar.cnr.it (accessed on 1 July 2018) .
43. Laugraud, B.; Piérard, S.; Van Droogenbroeck, M. LaBGen: A method based on motion detection for generating the background of a scene. *Pattern Recognit. Lett.* **2017**, *96*, 12–21. [CrossRef]
44. Laugraud, B.; Van Droogenbroeck, M. Is a Memoryless Motion Detection Truly Relevant for Background Generation with LaBGen? In Proceedings of the International Conference on Advanced Concepts for Intelligent Vision Systems, Antwerp, Belgium, 18–21 September 2017; pp. 443–454.
45. De Gregorio, M.; Giordano, M. Background estimation by weightless neural networks. *Pattern Recognit. Lett.* **2017**, *96*, 55–65. [CrossRef]
46. Prativadibhayankaram, S.; Luong, H.V.; Le, T.H.; Kaup, A. Compressive Online Robust Principal Component Analysis with Optical Flow for Video Foreground-Background Separation. In Proceedings of the Eighth International Symposium on Information and Communication Technology, Nha Trang, Vietnam, 7–8 December 2017; pp. 385–392.
47. Brand, M. Incremental Singular Value Decomposition of Uncertain Data with Missing Values. In Proceedings of the European Conference on Computer Vision, Copenhagen, Denmark, 28–31 May 2002.
48. Luong, H.V.; Seiler, J.; Kaup, A.; Forchhammer, S. Sparse Signal Reconstruction with Multiple Side Information using Adaptive Weights for Multiview Sources. In Proceedings of the IEEE International Conference on Image Processing, Phoenix, AZ, USA, 25–28 September 2016.
49. Luong, H.V. Available online: https://github.com/huynhlvd/corpca (accessed on 1 July 2018).
50. Beck, A.; Teboulle, M. A Fast Iterative Shrinkage-Thresholding Algorithm for Linear Inverse Problems. *SIAM J. Imaging Sci.* **2009**, *2*, 183–202. [CrossRef]
51. Cai, J.F.; Candès, E.J.; Shen, Z. A Singular Value Thresholding Algorithm for Matrix Completion. *SIAM J. Optim.* **2010**, *20*, 1956–1982. [CrossRef]
52. Szeliski, R. *Computer Vision: Algorithms and Applications*; Springer Science & Business Media: Berlin, Germany, 2010.
53. Li, L.; Huang, W.; Gu, I.Y.H.; Tian, Q. Statistical modeling of complex backgrounds for foreground object detection. *IEEE Trans. Image Process.* **2004**, *13*, 1459–1472. [CrossRef] [PubMed]
54. Dikmen, M.; Tsai, S.F.; Huang, T.S. Base selection in estimating sparse foreground in video. In Proceedings of the 16th IEEE International Conference on Image Processing (ICIP), Cairo, Egypt, 7–10 November 2009; pp. 3217–3220.
55. Cucchiara, R.; Grana, C.; Piccardi, M.; Prati, A. Detecting moving objects, ghosts, and shadows in video streams. *IEEE Trans. Pattern Anal. Mach. Intell.* **2003**, *25*, 1337–1342. [CrossRef]
56. Luong, H.V.; Prativadibhayankaram, S. Available online: https://github.com/huynhlvd/corpca-of (accessed on 1 July 2018).