Hindawi
Journal of Electrical and Computer Engineering
Volume 2019, Article ID 7675805, 15 pages
https://doi.org/10.1155/2019/7675805

Research Article

Panning and Jitter Invariant Incremental Principal Component Pursuit for Video Background Modeling

Gustavo Chau and Paul Rodríguez

Department of Electrical Engineering, Pontificia Universidad Católica del Perú, Lima, Peru

Correspondence should be addressed to Gustavo Chau; gustavo.chau@pucp.edu.pe

Received 20 July 2018; Accepted 2 September 2018; Published 3 February 2019

Academic Editor: Nicolas Younan

Video background modeling is an important preprocessing stage for various applications, and principal component pursuit (PCP) is among the state-of-the-art algorithms for this task. One of the main drawbacks of PCP is its sensitivity to jitter and camera movement. This problem has only been partially solved by a few methods devised for jitter or small transformations. However, such methods cannot handle the case of moving or panning cameras in an incremental fashion. In this paper, we greatly expand the results of our earlier work, in which we presented a novel, fully incremental PCP algorithm, named incPCP-PTI, which was able to cope with panning scenarios and jitter by continuously aligning the low-rank component to the current reference frame of the camera. To the best of our knowledge, incPCP-PTI is the first low-rank plus additive incremental matrix method capable of handling these scenarios in an incremental way. The results on synthetic videos and Moseg, DAVIS, and CDnet2014 datasets show that incPCP-PTI is able to maintain a good performance in the detection of moving objects even when panning and jitter are present in a video. Additionally, in most videos, incPCP-PTI obtains competitive or superior results compared to state-of-the-art batch methods.

1. Introduction

Video background modeling consists of segmenting the "foreground" or moving objects from the static "background." It is an important first step in various computer vision applications [1] such as abnormal event identification [2] and surveillance [3].

Several video background modeling methods, using different approaches such as Gaussian mixture models [4], kernel density estimations [5], or neural networks [6], exist in the literature. More comprehensive surveys of other methods are presented in [1, 7]. Principal component pursuit (PCP) is currently considered to be one of the leading algorithms for video background modeling [8]. Formally, PCP was introduced in [9] as the nonconvex optimization problem:

$$\begin{aligned} \arg\min_{L,S} \quad & \operatorname{rank}(L) + \lambda \|S\|_0, \\ \text{s.t.} \quad & D = L + S, \end{aligned} \tag{1}$$

where the matrix $D \in \mathbb{R}^{m \times n}$ is formed by the n observed frames, each of size $m = N_r \times N_c \times N_d$ (rows, columns, and number of channels, respectively); $L \in \mathbb{R}^{m \times n}$ is a low-rank matrix representing the background; $S \in \mathbb{R}^{m \times n}$ is a sparse matrix representing the foreground; λ is a fixed global regularization parameter; and $\operatorname{rank}(L)$ is the rank of L and $||S||_0$ is the ℓ_0 norm of S.

Although the convex relaxation is given by

$$\begin{aligned} \arg\min_{L,S} \quad & \|L\|_* + \lambda \|S\|_1, \\ \text{s.t.} \quad & D = L + S, \end{aligned} \tag{2}$$

where $\|L\|_*$ is the nuclear norm of matrix L (i.e., $\sum_k |\sigma_k(L)|$, the sum of the singular values of L) and $||S||_1$ is the ℓ_1 norm of S, which is at the core of most PCP algorithms (including the augmented Lagrange multiplier (ALM) and inexact ALM (iALM) algorithms [10, 11]), there exists several others (for a complete list, see [12], Table 4). In particular, we point out

$$\begin{aligned} \arg\min_{L,S} \quad & \frac{1}{2}\|L + S - D\|_F^2 + \lambda \|S\|_1, \\ \text{s.t.} \quad & \operatorname{rank}(L) \leq r, \end{aligned} \tag{3}$$

where $\|\cdot\|_F^2$ is the Frobenius norm, which was originally proposed in [13] since we will use it as the starting point of our proposed method (Sections 2.2.2 and 3.1).

Bouwmans and Zahzah [8] showed that PCP provides state-of-the-art performance in video background modeling problems but also states some of its limitations.

First, PCP is inherently a batch method with high computational and memory requirements. This problem has been addressed in the past by means of solutions based on rank-1 updates for thin SVD [14, 15] (applied to (3)), by low-rank subspace tracking [16] (applied to (2)), stochastic optimization [17] (which applies the maximum-margin matrix factorization (M3F) method [18] to (2)), or random sampling [19] (also applied to (2)).

The second shortcoming of PCP, which is particularly relevant to the present work, is its sensitivity to jitter and its inability to cope with panning video frames. For a general review and classification of methods for motion segmentation able to cope with different degrees of camera motion, we recommend [20] and the many references therein. Among the methods based on low-rank plus additive matrices model, we highlight the robust alignment by sparse and low-rank decomposition (RASL) method [21]. This method used (2) as its starting point and addressed the problem of jitter in PCP using a series of geometric transformations on the observed frame, but as originally casted, it is a batch method. On the contrary, t-GRASTA [22] and incPCP-TI [23], which used (2) and (3) apiece as their starting point, addressed the problem of jitter in a semi-incremental or fully incremental way by applying geometric transformation to the observed frames or low-rank component, respectively. Other proposed methods are robust against moving camera and panning [24–26], but all of them are batch or semibatch methods; furthermore, all of them used (2) as their starting point and also used the same general ideas (4) as RASL. Recently, Gao et al. [27] presented a new batch PCP method that produces a panoramic low-rank component that spans the entire field of view, which gives much better results in long panning sequences. We notice, nonetheless, that a fully online PCP algorithm able to cope with both jitter and panning is still an open problem. This phenomenon is of particular importance in some applications such as surveillance systems that use moving traffic or air cameras.

In the present study, we expand our previous work [28], where we proposed to address the panning problem by modifying the optimization problem solved by incPCP-TI [14], which in turn uses (3) as its starting point, and y applying a set of transformations to the low-rank component that are updated with each incoming new frame. We substantially expand [28] by

(1) expanding the theoretical basis of our algorithm so that the present manuscript is self-contained

(2) testing our algorithm in an additional real-life dataset with moving cameras

(3) comparing our algorithm with two previously proposed batch methods

Our computational experiments on synthetically created datasets and publicly available videos of the Moseg [29], DAVIS [30], and CDnet2014 [31] datasets show that the proposed algorithm, henceforth referred as panning and transformation invariant incPCP (incPCP-PTI), is able to correctly handle video background modeling in panning and basic jitter conditions.

2. Previous Related Work

2.1. Batch Methods. In this section, two previous motion segmentation batch methods that work under jitter/panning conditions are reviewed. For a more complete review of all available methods, refer [20]. The two algorithms hereby described work on batch fashion but were chosen as a comparison benchmark in this paper due to the public availability of their codes and/or binary executables. It is noted that although [27] recently published a new PCP method for moving cameras, the algorithm was not chosen for comparison due to the unavailability of public code or executables.

2.1.1. Segmentation by Long-Term Video Analysis. In [29], the authors proposed to use a dense point tracker based on variational optical flow in which, instead of the classical two-frame approach of optical flow, long-term analysis is used. It is worth mentioning that following the general classification of methods for motion segmentation proposed in [12, 29] was catalogued as a trajectory classification method (see Table 3 of [12] for a summary of the aforementioned classification, along with their associated properties). After the initial tracking of points, spectral clustering with a spatial regularity constraint is utilized to form groups of point trajectories corresponding to different objects in the image. Finally, an energy minimization model is used to transform the clusters into a dense segmentation of moving objects. Throughout this paper, this method will be referred as LTVA.

2.1.2. DECOLOR. The detecting contiguous outliers in the low-rank representation (DECOLOR) method [25] uses a nonconvex penalty and a Markov random field [32] model to detect outliers that correspond to moving objects. Bouwmans et al. [12] classified this algorithm as a low rank and as sparse representation method [21]. For moving cameras, the method uses a transformation obtained from a prealignment to the middle frame. The prealignment is performed using the robust multiresolution method proposed in [33] and DECOLOR then iteratively refines this transformation.

2.2. Online or Semionline Methods. In this section, two previous online or partially online PCP methods that work under jitter conditions are reviewed. It should be noted that, without modification, these two methods are not directly applicable to panning scenarios.

2.2.1. t-GRASTA. The Grassmannian Robust Adaptive Subspace Tracking Algorithm (GRASTA) [16] is a

semionline method for low-rank subspace tracking that has been applied to the foreground-background separation problem. GRASTA is not a fully online algorithm as it requires an initialization stage to obtain an initial low-rank subspace from the first p frames. A modification called t-GRASTA was presented in [22], and it is based on the Robust Alignment by Sparse and Low-Rank decomposition (RASL) algorithm [21]. RASL tries to handle the misalignment in the video frames by solving

$$\begin{aligned} &\arg\min_{L,S,\tau} && \|L\|_* + \lambda\|S\|_1, \\ &\text{s.t.} && \tau(D) = L + S, \end{aligned} \tag{4}$$

where $\tau(\cdot)$ are a series of per frame transformations that align all the observed frames; it is straightforward to note that (4) is an extension of (2).

The non-linearity in the transformations τ of (4) is handled via a linearization using the Jacobian. The main drawback of t-GRASTA is that, aside from the required low-rank subspace initialization, the initial transformation τ is estimated by using a similarity transformation obtained from a series of three points manually chosen from each of the p initial frames. This initialization stage severely constraints its application in automatic processes and reduces its applicability in panning scenarios, as the feature points in initial frames may not be present on subsequent frames.

2.2.2. incPCP-TI. The incPCP-TI [23] considers the optimization problem

$$\begin{aligned} &\arg\min_{L^*,S,\mathcal{T}} && \frac{1}{2}\left\|D - \mathcal{T}(L^*) - S\right\|_2^2 + \lambda||S||_1, \\ &\text{s.t.} && \operatorname{rank}(L^*) \le r, \end{aligned} \tag{5}$$

where D is the observed video sequence that suffers from jitter, L^* is the properly aligned low-rank representation, and $\mathcal{T} = \{\mathcal{T}_k\}$ is a set of transformations that compensate translational and rotational jitter; that is,

$$D = \mathcal{T}(D^*) = H * R(D, \alpha), \tag{6}$$

where D^* represents the unobserved jitter-free video sequence, $H = \{h_k\}$ is a set of filters that independently models translation for each frame, $*$ represents the convolution, and $R(D, \alpha)$ is a set of independent rotations applied to each frame with angle $\alpha = \{\alpha_k\}$. It is interesting to note that $\mathcal{T} = \tau^{-1}$; that is, the transformation used in (5) can be understood as the inverse of the transformation used in RASL or t-GRATSA (4).

In [14, 15], a computationally efficient and fully incremnetal algorithm, based on rank-1 updates for thin SVD [34–37] (also see Section 2.3), was proposed to solve (3); in [23], it was shown that, since (5) is based on (3), such incremental solution can also be used: letting $\mathbf{d}_k$, $k \in \{1, 2, \ldots, n\}$ represent each frame of the observed video D, and using similar relationships for $\mathbf{s}_k$ and $\mathbf{l}_k^*$ w.r.t. S and L^*, respectively, then indeed the solution of

$$\begin{aligned} &\arg\min_{L,S,H,\alpha} && \frac{1}{2}\sum_k \left\|h_k * R(l_k^*, \alpha_k) + s_k - d_k\right\|_{\mathrm{F}}^2 + \lambda||S||_1 + \gamma\sum_k \left|\left|h_k\right|\right|_1, \\ &\text{s.t.} && \operatorname{rank}(L^*) \le r, \end{aligned} \tag{7}$$

can be efficiently computed in an incremental fashion ([23], Section 3.3 for details).

2.3. Incremental and Rank-1 Modifications for Thin SVD. Given a matrix $D \in \mathbb{R}^{m\times l}$ with thin SVD $D = U_0\Sigma_0 V_0^T$ where $\Sigma_0 \in \mathbb{R}^{r\times r}$ and column vectors $\mathbf{a}$ and $\mathbf{b}$ (with m and l elements, respectively), note that

$$D + \mathbf{a}\mathbf{b}^T = \begin{bmatrix} U_0 & \mathbf{a} \end{bmatrix} \begin{bmatrix} \Sigma_0 & \mathbf{0} \\ \mathbf{0}^T & 1 \end{bmatrix} \begin{bmatrix} V_0 & \mathbf{b} \end{bmatrix}^T, \tag{8}$$

where $\mathbf{0}$ is a zero column vector of the appropriate size. Based on [35, 36], as well as on [37], we will briefly describe an incremental (thin) SVD and rank-1 modifications (update, downdate, and replace) for thin SVD.

The generic operation consisting of the Gram–Schmidt orthonormalization of $\mathbf{a}$ and $\mathbf{b}$ w.r.t. U_0 and V_0, i.e., $\mathbf{x} = U_0^T\mathbf{a}$, $\mathbf{z}_x = \mathbf{a} - U\mathbf{x}$, $\rho_x = ||\mathbf{z}_x||_2$, $\mathbf{p} = (1/\rho_x)\mathbf{z}_x$ and $\mathbf{y} = V_0^T\mathbf{b}$, $\mathbf{z}_y = \mathbf{b} - V\mathbf{y}$, $\rho_y = \|\mathbf{z}_y\|_2$, and $\mathbf{q} = (1/\rho_y)\mathbf{z}_y$, is used as a first step for all the cases described below.

2.3.1. Incremental or Update Thin SVD. Given $\mathbf{d} \in \mathbb{R}^{m\times 1}$, we want to compute thin $\operatorname{SVD}(\begin{bmatrix} D & \mathbf{d} \end{bmatrix}) = U_1\Sigma_1 V_1^T$, with (i) $\Sigma_1 \in \mathbb{R}^{r+1\times r+1}$ or (ii) $\Sigma_1 \in \mathbb{R}^{r\times r}$. In this case, we note that $\begin{bmatrix} D & \mathbf{0} \end{bmatrix} = U_0\Sigma_0\begin{bmatrix} V_0 & \mathbf{0} \end{bmatrix}^T$ and that $\begin{bmatrix} D & \mathbf{d} \end{bmatrix} = \begin{bmatrix} D & \mathbf{0} \end{bmatrix} + \mathbf{d}\mathbf{e}^T$, where $\mathbf{e}$ is a unit vector (with $l+1$ elements in this case); then, (8) is equivalent to (9) and (10), where $\widehat{\Sigma} \in \mathbb{R}^{r+1\times r+1}$:

$$\begin{bmatrix} D & \mathbf{0} \end{bmatrix} + \mathbf{d}\mathbf{e}^T = \begin{bmatrix} U_0 & \mathbf{p} \end{bmatrix} \cdot \left(G\widehat{\Sigma}H^T\right) \cdot \begin{bmatrix} V_0^T & \mathbf{0} \\ \mathbf{0}^T & 1 \end{bmatrix}, \tag{9}$$

$$G\widehat{\Sigma}H^T = \operatorname{SVD}\left(\begin{bmatrix} \Sigma_0 & \mathbf{x} \\ \mathbf{0}^T & \rho_x \end{bmatrix}\right). \tag{10}$$

Using (11), we get $\operatorname{SVD}(\begin{bmatrix} D & \mathbf{d} \end{bmatrix})$ with (i) $\Sigma_1 \in \mathbb{R}^{r+1\times r+1}$; similarly using (12), we get $\operatorname{SVD}(\begin{bmatrix} D & \mathbf{d} \end{bmatrix})$ with (ii) $\Sigma_1 \in \mathbb{R}^{r\times r}$ (Matlab notation is used to indicate array slicing operations):

$$\begin{aligned} U_1 &= \begin{bmatrix} U_0 & \mathbf{p} \end{bmatrix} \cdot G, \\ \Sigma_1 &= \widehat{\Sigma}, \\ V_1 &= \begin{bmatrix} V_0 & \mathbf{0} \\ \mathbf{0}^T & 1 \end{bmatrix} \cdot H, \end{aligned} \tag{11}$$

$$\begin{aligned} U_1 &= U_0 \cdot G(1:r, 1:r) + \mathbf{p} \cdot G(r+1, 1:r), \\ \Sigma_1 &= \widehat{\Sigma}(1:r, 1:r), \\ V_1 &= \left[V_0 \cdot H(1:r, 1:r); H(r+1, 1:r)\right]. \end{aligned} \tag{12}$$

2.3.2. Downdate Thin SVD. Given $\begin{bmatrix} D & \mathbf{d} \end{bmatrix} = U_0\Sigma_0 V_0^T$, with $\Sigma_0 \in \mathbb{R}^{r\times r}$, we want to compute thin $\operatorname{SVD}(D) = U_1\Sigma_1 V_1^T$

with r singular values. Noting that $\left[\begin{array}{cc} D & \mathbf{0} \end{array}\right] = \left[\begin{array}{cc} D & \mathbf{d} \end{array}\right] + (-\mathbf{d})\mathbf{e}^T$, then the rank-1 modification (8) is equivalent to

$$\left[\begin{array}{cc} D & \mathbf{d} \end{array}\right] + (-\mathbf{d})\mathbf{e}^T = \left[\begin{array}{cc} U_0 & \mathbf{0} \end{array}\right] \cdot \left(G\widehat{\Sigma}H^T\right) \cdot \left[\begin{array}{cc} V_0 & \mathbf{q} \end{array}\right]^T,$$
$$G\widehat{\Sigma}H^T = \mathrm{SVD}\left(\left[\begin{array}{cc} \Sigma_0 - \Sigma_0\mathbf{y}\mathbf{y}^T & -\rho_y \cdot \Sigma_0\mathbf{y} \\ 0^T & 0 \end{array}\right]\right), \tag{13}$$

from which we can compute thin SVD(D) via the following equation:

$$\begin{aligned} U_1 &= U_0 \cdot G(1:r, 1:r), \\ \Sigma_1 &= \widehat{\Sigma}(1:r, 1:r), \\ V_1 &= V_0 \cdot H(1:r, 1:r) + \mathbf{q} \cdot H(r+1, 1:r). \end{aligned} \tag{14}$$

2.3.3. Replace Thin SVD. Given $\left[\begin{array}{cc} D & \mathbf{d} \end{array}\right] = U_0\Sigma_0V_0^T$, with $\Sigma_0 \in \mathbb{R}^{r\times r}$, we want to compute thin SVD$\left(\left[\begin{array}{cc} D & \widehat{\mathbf{d}} \end{array}\right]\right) = U_1\Sigma_1V_1^T$ with r singular values. This case can be understood as a mixture of the previous cases and can be easily derived noticing that $\left[\begin{array}{cc} D & \widehat{\mathbf{d}} \end{array}\right] = \left[\begin{array}{cc} D & \mathbf{d} \end{array}\right] + (\widehat{\mathbf{d}} - \mathbf{d})\mathbf{e}^T$.

Finally, we point out that the computational complexity of any of the above procedures ([35], Section 3 and [37], Section 4) is upper bounded by $O(10 \cdot m \cdot r) + O(r^3) + O(3 \cdot r \cdot l)$. If $r \ll m, l$ holds, then the complexity is dominated by $O(10 \cdot m \cdot r)$.

3. Methods

3.1. Proposed incPCP-PTI Method. The proposed algorithm (named incPCP-PTI) is a modification of the previously proposed incPCP-TI [23] so that it is able to handle panning and camera motion. It was briefly presented in [28], and is more thoroughly explained and evaluated in this work. The method continuously estimates the alignment transformation $\mathcal{T}$ so that $\mathcal{T}(\mathbf{l}_k^*) = \mathbf{d}_k$, i.e., the transformation that aligns the previous low-rank representation with the observed current frame. Thus, incPCP-PTI effectively uses $\mathcal{T}(\mathbf{l}_k^*)$ as a local estimation of a composite panoramic background image. After applying such transformation to L^*, the PCP problem can be solved in the reference frame of $\mathbf{d}_k$. After this initial alignment, it is considered that only minor jitter remains in the image and so a procedure similar to incPCP-TI is utilized by estimating a transformation ξ_k for the k-th frame. However, instead of solving the Affinely Constrained Matrix Rank Minimization [38] as in the original incPCP-TI [14], the low-rank approximation problem is solved in the reference frame of $\mathbf{d}_k$ by applying ξ_k^{-1} to the residual $\mathbf{d}_k - \mathbf{s}_k$. The whole procedure is presented in Algorithm 1. This algorithm makes use of the incSVD, repSVD, and downSVD operators, which correspond to the thin SVD update, replacement, and downdate operators, respectively (Section 2.3).

In line 3 of Algorithm 1, the latest low-rank frame $\mathbf{l}_k$ is aligned to the current frame $\mathbf{d}_k$. The transformation is estimated as the composition of a translation and rotation. Such found aligned transformation $\mathcal{T}_k(L)$ is used then to update the whole low-rank matrix representation L to the current reference axis (lines 4 and 5 of Algorithm 1) in order to obtain L^*. After this initial align transformation is performed, it is assumed that only minor misalignments, modeled by ξ_k, due to jitter remain (line 10 of Algorithm 1).

The ghosting suppression mentioned in line 16 is detailed in Section 3.3. The shrinkage in line 9 of Algorithm 1 can be performed by either soft thresholding or projection on the ℓ_1 ball. Soft thresholding is performed with a simple element-wise shrinkage operator (shrink$(x, \lambda) =$ sign(x)max$(0, |x| - \lambda)$). Projection onto the ℓ_1 ball is detailed in Section 3.2. For all our experiments, the latter was chosen.

3.2. Projection on the ℓ_1 Ball. Although theoretical guidance is available for selecting a minimax optimal regularization parameter λ in (2) [39], practical problems do not fully satisfy the idealized assumptions, and thus λ often has to be heuristically tuned. This problem is also observed if (3) is used instead of (2).

To tackle this problem, Rodrıguez and Wohlberg [40] introduced the alternative relaxation of (1) given by

$$\begin{aligned} \arg\min_{L,S} \quad & \frac{1}{2}\|L + S - D\|_{\mathrm{F}}^2, \\ \text{s.t.} \quad & ||S||_1 \le \mu, \ \mathrm{rank}(L) \le r, \end{aligned} \tag{15}$$

which can also be incrementally solved via rank-1 updates for thin SVD (as is the case of the incPCP and related algorithms [14, 15, 23]); however, (15) has the advantage that a simple heuristic can be derived for the adaptive selection of μ for each frame. Furthermore, μ can be spatially adapted in order to reduce ghosting effects. The algorithm they propose is very similar to incPCP, save for the shrinkage step, which is calculated as $\mathbf{s}_k = \mathrm{proj}_{\|\cdot\|_1}(\mathbf{d}_k - \mathbf{l}_k, \mu)$, where

$$\begin{aligned} \mathrm{proj}_{\|\cdot\|_1}(\mathbf{u}, \mu) \triangleq \min_{\mathbf{x}} \quad & \frac{1}{2}||\mathbf{x} - \mathbf{u}||_2^2, \\ \text{s.t.} \quad & \|\mathbf{x}\|_1 \le \mu. \end{aligned} \tag{16}$$

Thus, for the shrinkage step, the solution is given by projections into the ℓ_1 ball of radius μ.

While there are several well-known and efficient algorithms that solve (16), studies [40–43] used the algorithm in [44], a recently published algorithm, for solving (16) that has a better computational performance than either that in [41] or [42].

Furthermore, Rodrıguez and Wohlberg [40] also proposed a simple scheme for adapting μ_k with every frame, which is given by

$$\mu_k = \alpha \cdot \left\|\mathbf{d}_k - \mathbf{l}_k\right\|_1, \tag{17}$$

where α is a value between 0.5 and 0.75.

3.3. Ghosting Suppression. Ghosting refers to when the foreground estimates include phantoms or smear replicas from actual moving objects. Rodríguez and Wohlberg [45] proposed a procedure for ghosting suppression in the

Input: observed video D, internal parameters for shrinkage, internal parameters for transformation estimation, number of innerLoops iL, background frames bl, $m = k_0$
Initialization: $L + S = D(:, 1 : k_0)$, initial rank r, $[U_r, \Sigma_r, V_r] = \text{partialSVD}(L, r)$
(1) **for** $k = k_0 + 1 : n$ **do**
(2) $++m$;
(3) find T_k such that $\|T_k(\mathbf{l}_{k-1}) - \mathbf{d}_k\|_2$ is minimized
(4) obtain $L^* = T_k(L) = [T_k(\mathbf{l}_1), \ldots, T_k(\mathbf{l}_{k-1})]$
(5) $[U_k, \Sigma_k, V_k] = \text{partialSVD}(L^*, r)$
(6) $[U_k, \Sigma_k, V_k] = \text{incSVD}(d_k, U_k, \Sigma_k, V_k)$
(7) **for** $j = 1 : iL$ **do**
(8) $\mathbf{l}_k^* = U_k(:, 1 : r) * \Sigma_k * (V_k(\text{end}, :)')$
(9) $\mathbf{s}_k = \text{shrink}(\mathbf{d}_k - \mathbf{l}_k^*)$
(10) $\xi_k = \text{argmin}_\xi \|\xi(\mathbf{l}_k^*) - (\mathbf{d}_k - \mathbf{s}_k)\|_2$
(11) **if** $j == iL$ **then**
(12) break
(13) $\rho = \xi_k^{-1}(\mathbf{d}_k - \mathbf{s}_k)$
(14) $[U_k, \Sigma_k, V_k] = \text{repSVD}(\mathbf{d}_k, \rho, U_k, \Sigma_k, V_k)$
(15) **end**
(16) Apply ghosting suppression
(17) **if** $m \geq bL$ **then**
(18) $\text{downSVD}(1\text{stcolumn}, U_k, \Sigma_k, V_k)$
(19) Update k if necessary
(20) **end**

Algorithm 1: IncPCP-PTI.

incPCP algorithm which consists using binary masks obtained from different frames in order to remove the ghosts from the low-rank component. In this approach, two sparse components at different time steps n_1 and n_2 are used to compute respective binary masks $\mathbf{m}_k^{(n_1)}$ and $\mathbf{m}_k^{(n_2)}$. These masks will include the moving objects as well as ghosts. A new binary mask $\mathbf{b}_k = (\mathbf{m}_k^{(n_1)} \cap \mathbf{m}_k^{(n_2)})^C$, i.e., the complement of the intersection of binary masks obtained from the aforementioned two frames, will include, with high probability, all pixels of the background that are not occluded by a moving object. $\mathbf{b}_k$ can then be used to generate a modified input frame $\widehat{d}_k^{(n)} = \mathbf{d}_k \odot \mathbf{b}_k + \mathbf{l}_k \odot (1 - \mathbf{b}_k)$, where $\odot$ represents an Hadamard product, which is used to update the low-rank component. Additionally, if the procedure with the ℓ_1 ball projection described in Section 3.2 is used for the shrinkage step, μ_k can be spatially adapted in order to reduce ghosting [40]. Based on the difference between current and previous sparse approximation $\mathbf{z}_k = \mathbf{s}_k - \mathbf{s}_{k-1}$, a binary mask $\mathbf{m}_k$ can be computed and then the sparse component is modified as

$$\mathbf{s}_k = \left(1 - \mathbf{m}_k\right) \odot \widehat{s}_k + \mathbf{m}_k \odot \widetilde{s}_k, \tag{18}$$

where $\widehat{s}_k = \text{proj}_{\|\cdot\|_1}(\mathbf{d}_k - \mathbf{l}_k, \mu_k)$ and $\widetilde{s}_k = \text{proj}_{\|\cdot\|_1}(\mathbf{m}_k \odot \widehat{s}_k, \mu_k^{(g)})$ and $\mu_k^{(g)} = \beta \cdot \|\mathbf{m}_k \odot \widehat{s}\|_k 1$, where $\text{prox}_{\|\cdot\|_1}(\cdot)$ is defined in (16), and β is suggested to take values between 0.1 and 0.3.

4. Description of Datasets and Computational Experiments

For the evaluation of the proposed incPCP-PTI algorithm, four datasets were considered. The first dataset consists of synthetic jitter and panning videos. The second one consists of videos of real panning taken from the MoSeg dataset [29]. The third dataset consists of videos of the recently published DAVIS dataset [30, 46]. The last one was obtained from the CDnet2014 dataset [31]. All datasets are detailed in this section. All tests were carried out using GPU-enabled Matlab code running on an Intel i7-2600K CPU (8 cores, 3.40 GHz, 8 MB cache, and 32 GB RAM) with a 12 GB NVIDIA Tesla K40C GPU card. To the best of our knowledge, no other incremental low-rank plus additive matrix video background modeling technique capable of handling panning has been reported in the literature. This situation puts some constraints in our evaluations, and for most of our tests, we do comparisons with batch methods. Specific details for each of the datasets are described in their corresponding sections. Furthermore, to test the stability of incPCP-PTI to jitter, we also use stab + incPCP-PTI, which consists on a preprocessing stage using a recent state-of-the-art video stabilization technique [47] followed by incPCP-PTI.

4.1. Synthetic Datasets. A dataset with synthetic panning and jitter was generated from the 3rd Tower video of the USC Neovision2 dataset [48], which consists of 900 frames of size 1920×1088 pixel at 25 fps. For this purpose, a subregion of 720×480 pixel was selected from each frame and the centroid of the subregion was translated with each new frame in order to simulate an aerial panning scenario using the piecewise linear trajectory $u[n]$ given by

$$u[n] = \begin{cases} u[n-1] + v \cdot (1, 1), u[n-1]_x < Q, \\ u[n-1] + v \cdot (1, 0), u[n-1]_x \geq Q, \end{cases} \tag{19}$$

where $u[0]$ is the initial point (in this case, it is chosen as $(150, 688)$ pixel) and is the point of slope change in the curve (chosen as $Q = 500$ pixel). This process is depicted in Figure 1.

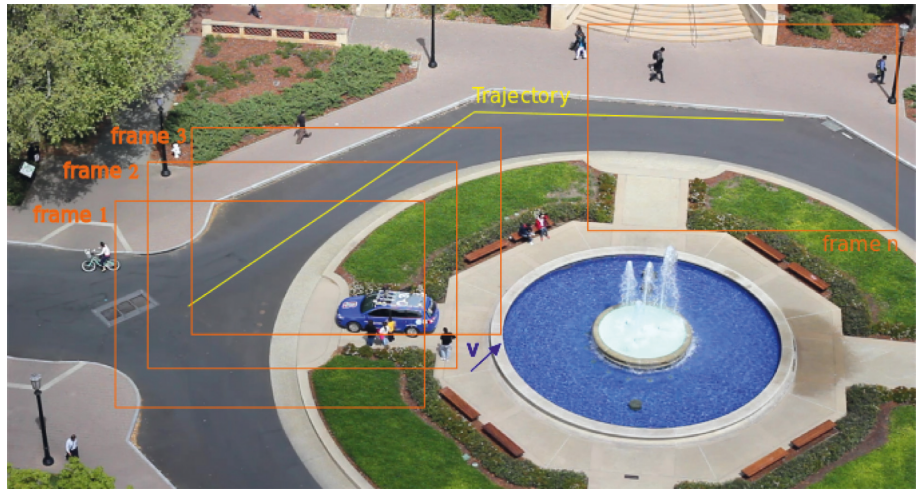

Figure 1: Construction of the synthetic panning and jitter dataset. The selected region (blue rectangles) was of 720 × 480 pixel, and the centroid of the region was translated at a velocity v (red vector) along a piecewise linear trajectory (green). This figure is a slightly modified version of Figure 1 in [28].

The panning velocity v was taken as 1, 3, and 5 pixels per frame. A fourth case in which the velocity changed randomly between 1 and 7 pixels per frame was also considered. This dataset will be referred as SP ("synthetic panning") dataset. Additionally, this same procedure was used to construct a dataset on jittered versions of the original frames. Each frame of the 3rd Tower video was jittered with random uniformly distributed translations on the $[-10, 10]$ pixels range and random uniformly distributed rotations on the $[-0.5, 0.5]$ degrees range. The same trajectory and subregion selection of the SP dataset was used. This second synthetic dataset will be referred as SPJ ("synthetic panning and jitter dataset") dataset. For both SP and SPJ datasets, the sparse approximation via the batch iALM method [10] using 20 outer iterations was used as a proxy ground truth by selecting the same regions that were selected from the original frames. The iALM was chosen as the proxy ground truth since, as reported in [8], Tables 6 and 7, its segmentation is considered to be reliable. For these synthetic datasets, the performance of the proposed algorithm was measured in terms of the normalized ℓ_1 distance:

$$M(\mathbf{s}_k) = \frac{\left\|\mathbf{s}_k^{\mathrm{gt}} - \mathbf{s}_k\right\|_1}{N}, \tag{20}$$

where $\mathbf{s}_k^{\mathrm{gt}}$ and $\mathbf{s}_k$ are the ground truth and computed sparse components for frame k, respectively, and N is the number of pixels of the frame. Considering images normalized between 0 and 1, the value of $M(\mathbf{s}_k)$ varies from 0 (perfect match with the ground truth) to 1.

For the SP dataset, only the incPCP-PTI method was evaluated. For the SPJ dataset, we evaluated incPCP-PTI and, as mentioned in Section 4, stab + incPCP-PTI, which consists of a preprocessing stage using a recent state-of-the-art video stabilization technique [47] followed by incPCP-PTI. This comparison had as objective determining if jitter is handled correctly by incPCP-PTI [23] alone. Additionally, we include a baseline comparison with the sparse components obtained with incPCP on the full Neovision2 Tower video and then segmented using the same procedure described in Section 4.1.

4.2. Moseg Dataset. We used 15 video sequences of the Freiburg-Berkeley Motion Segmentation (Moseg) Dataset [29, 49, 50]. We selected sequences that contained panning or camera movement. For all incPCP-PTI variants, three inner loops and a window size of 30 background frames were used. For the ℓ_1 ball projection, α was set to 0.75 and the ghosting suppression $n_2 - n_1$ was set to 20 frames. α controls the adaptation of τ, with lower α forcing a sparser solution, whereas the difference $n_2 - n_1$ controls the number of frames used for ghosting suppression. The binary mask obtained with incPCP-PTI was postprocessed using the computation of the convex hull of the connected objects [51]. It is noted that the application of this postprocessing was not applied to the other methods as it tended to reduce their performance.

For comparison, both LTVA and DECOLOR algorithms were used as a reference. The LTVA code was obtained from [52], and the default parameter of a spatial subsampling of 8 for the point tracking was used. The tracking component of the algorithm runs in single-threaded C, whereas the dense clustering component runs in CUDA C 5.5. The Single-threaded Matlab DECOLOR code was obtained from [53], and a tolerance of 1E-4 was used. All other parameters were left at their default values. For reporting the results, we further subdivided the selected videos into two categories: *short panning* (comprising nine *cars* sequences and one *people* sequences) and *long panning* (comprising 5 *marple* videos). In the former category, the final and first frames in the panning motion still share some common area, while in the latter, these two frames do not. This subdivision was necessary as DECOLOR performs a preregistration and was not able to work properly on the *long panning* sequences.

For all videos, the binary masks of the methods were compared to the ground truth provided in the dataset in order to obtain an F measure, defined as

$$F = \frac{2 \cdot P \cdot R}{P + R},$$
$$P = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}}, \quad (21)$$
$$R = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}},$$

where P and R stands for precision and recall, respectively, and TP, FN, and FP are the number of true positives, false negatives, and false positive pixels, respectively. It is noted that approximately only one in ten frames possessed ground truth information.

4.3. DAVIS Dataset. We used 10 video sequences of the DAVIS [30, 46]. We selected some sequences that contained panning or camera movement and that contained at least 50 frames. The algorithms were configured with the same parameters specified in Section 4.2 and the same F measure comparison is performed. However, as mentioned before, DECOLOR is not able to run on all sequences and just the F measures where its prealigment phase runs correctly are reported. Additionally, in this dataset, LTVA presented oversegmentation of some objects and, accordingly, two methods to obtain the final binary mask of moving objects are considered (this problem was not present in the Moseg sequences). The first, reported as LTVA-aPP (automatic postprocessing), considers an automatic selection of the mask by designating the object label with the largest area as the background and considering all others labels as foreground objects. The second, reported as LTVA-mPP (manual postprocessing), entails the manual selection of the labels corresponding to moving objects. Although the latter method is more accurate, for an automatic pipeline, LTVA-aPP would be the more realistic option.

4.4. Change Detection 2014 (CDnet2014) Dataset. Two videos from the PTZ category of the CDnet2014 dataset [31] were chosen:

(i) continuousPan(CP): 704 × 480 pixel, 1700 frames-color video containing a continuous panning of a PTZ camera. The video is almost jitter free.

(ii) intermittentPan(IP): 560 × 368 pixel, 3500 frame color of a PTZ camera that changes between two fixed positions. The video contains intermittent panning and additional real jitter.

As mentioned in the previous section, DECOLOR is unable to work on this type of long panning sequences and accordingly, only incPCP-PTI and LTVA are compared. The F measure was evaluated only on frames that contained ground truth motion. LTVA presented the same segmentation problems as in the DAVIS dataset. However, in this case, the mask did not provide a segmentation good enough to produce a manual postprocessing, and thus, only the results of LTVA + aPP are presented. All the parameters for incPCP-PTI and LTVA are the same as in previous sections. We also included a comparison with the edge based foreground background segmentation and interior classification (EFIC) [54] and its color version, C-EFIC [55]. These methods were chosen as they obtained the second and third best F measure in the PTZ category of the CDnet2014 dataset results [56]. The top performer in the category was not selected as it corresponded to a supervised convolutional neural network that needs proper training before classification. Unfortunately, no open code is available for EFIC and C-EFIC, and we only had access to the segmented binary masks submitted to the challenge [57, 58]. Due to this limitation, only a referential F measure could be computed. The absence of open code makes it difficult to ascertain if EFIC and C-EFIC can be implemented in a fully incremental way and to compare them in terms of computational performance. Additionally, as EFIC and C-EFIC already include a postprocessing step on the binary mask, we did not apply the convex hull postprocessing of the connected objects [51] that was used for incPCP-PTI.

5. Results

5.1. Synthetic Datasets

5.1.1. SP Dataset. The distance $M(\mathbf{s}_k)$ (20) computed for each frame of the different videos of the SP dataset is shown in Figure 2. Table 1 shows the average distance $\overline{M(\mathbf{s}_k)}$ and average time for processing one frame along with a baseline metric, described in Section 4.1. It can be noticed that the distance tends to increase as the panning velocity increases but the distance in all cases maintains relatively small (below 0.01).

5.1.2. SPJ Dataset. Representative frames of the SPJ video with changing velocity and the segmented sparse components with incPCP-PTI and stab + incPCP-PTI are shown in Figure 3. The distance $M(\mathbf{s}_k)$ computed for each frame of the different videos of the SPJ dataset are shown in Figures 4 and 5 for incPCP-PTI and stab + incPCP-PTI.

5.2. Moseg Dataset. Representative frames of the video and the segmented sparse components for the *cars8*, *people1*, and *marple13* videos of the Moseg dataset are shown in Figure 6. Tables 2 and 3 show the F measure obtained in the short and long panning subsets, respectively. As noted in the previous sections, DECOLOR did not properly work on the long panning sequences, and so it is excluded from the comparisons in this subset.

5.3. DAVIS Dataset. Representative frames of the video and the segmented sparse components for the *tennis*, *horsejump-high*, *swing*, and *dog-gooses* videos of the Davis dataset are shown in Figure 7. Table 4 shows the F measure obtained in the short and long panning subsets, respectively. The sequences in which DECOLOR did not run properly were left unreported. As described in Section 4.3, LTVA-aPP and

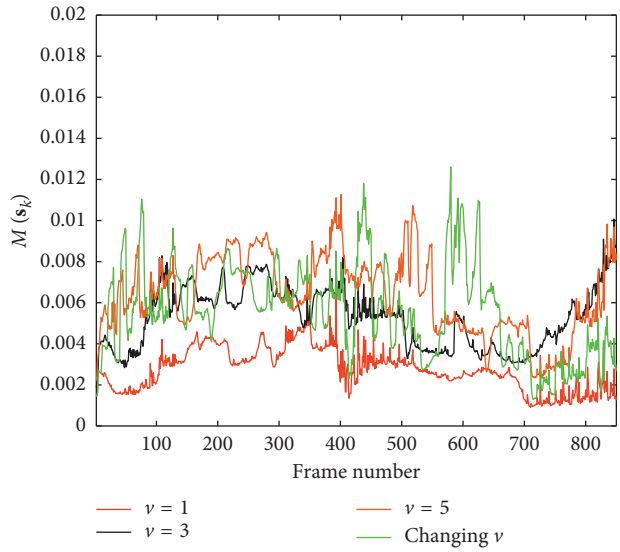

Figure 2: Value of distance δ between binary mask and ground truth frames for incPCP-PTI on the SP dataset (Section 4.1). This figure is a slightly modified version of Figure 2 in [28].

Table 1: Value of average distance $\overline{M(\mathbf{s}_k)}$ between binary mask and ground truth frames for incPCP-PTI and baseline incPCP on the SP dataset (Section 4.1). Computational time per frame for incPCP-PTI is also shown.

Dataset	IncPCP-PTI $\overline{M(\mathbf{s}_k)}$	IncPCP-PTI average time per frame (seconds)	Baseline incPCP $\overline{M(\mathbf{s}_k)}$
$v = 1$	0.0028	2.10	0.0024
$v = 3$	0.0053	2.10	0.0047
$v = 5$	0.0064	2.11	0.0054
Changing v	0.0055	2.09	0.0029

LTVA-mPP refer to automatic and manual postprocessing of the binary masks generated by the LTVA method, respectively.

5.4. CDnet2014 Dataset. Representative frames of the video and the segmented sparse components for the CP and IP videos are shown in Figures 8 and 9, respectively. Figure 10 shows the F measure (with no postprocessing) for incPCP-PTI (grayscale and color versions) and EFIC and C-EFIC on the frames of the CP video, while Figure 11 shows the same metric for all methods on the frames of the IP video. Tables 5 and 6 show the average F measure and computational time obtained overall frames. For stab + incPCP-PTI, the computational time is shown as (total stabilization time) + (incPCP-PTI time per frame). For LTVA, the total time of the batch execution was divided over the total number of frames in order to obtain an average time per frame.

6. Discussion

It is observed in the results of Section 5.1.1 that, as expected, the distance $M(\mathbf{s}_k)$ increased; that is, the sparse approximation was worse, as the panning velocity increased. On the contrary, incPCP-PTI is able to maintain an adequate performance even when the panning velocity changes. Also expected is the fact that adding jitter to the panning scenario (Section 5.1.2) increased the distance $M(\mathbf{s}_k)$ for all panning velocities with respect to their jitter-free counterparts. The overall stability of the estimated distance also decreased, as evidenced in the higher variability of the curves in Figure 4. The inclusion of a video stabilization preprocessing technique (stab + incPCP-PTI) seemed to decrease such variability, as evidenced in Figure 5. Nevertheless, even with jitter, standalone incPCP-PTI maintained a low average $M(\mathbf{s}_k)$ distance and its performance is comparable with stab + incPCP-PTI, as can be observed in Table 7. Furthermore, although incPCP-PTI obtained higher distances than baseline incPCP, values tend to be close to each other and, for all tested velocities, incPCP-PTI managed to maintain a very small distance from the ground truth (below 0.01 for all cases).

The results of Table 2 (related to the Moseg dataset, Section 4.2) suggest that incPCP-PTI is able to perform comparably to DECOLOR, even though the latter is a batch method and our proposed method is incremental. LTVA has substantially higher average F measure for this particular dataset, although working in a batch fashion. In Table 3, the same trend is observed. As mentioned above, DECOLOR has problems working in these sequences due to its prealignment phase failing to find a suitable unique frame for reference. The low performance of incPCP-PTI in some of the Moseg sequences might stem from the short number of video frames that cause the initial low-rank estimation of PCP to be less precise.

The results of Sections 5.3 and 5.4 (related to the DAVIS and CDNet datasets, respectively, Sections 4.3 and 4.4) suggest that incPCP-PTI can perform adequately in longer real panning videos with more complex scenarios. Regarding the DAVIS dataset, we observe that the highest performance is obtained with LTVA-mPP. Nevertheless, this is a batch method and the final binary segmentation required human

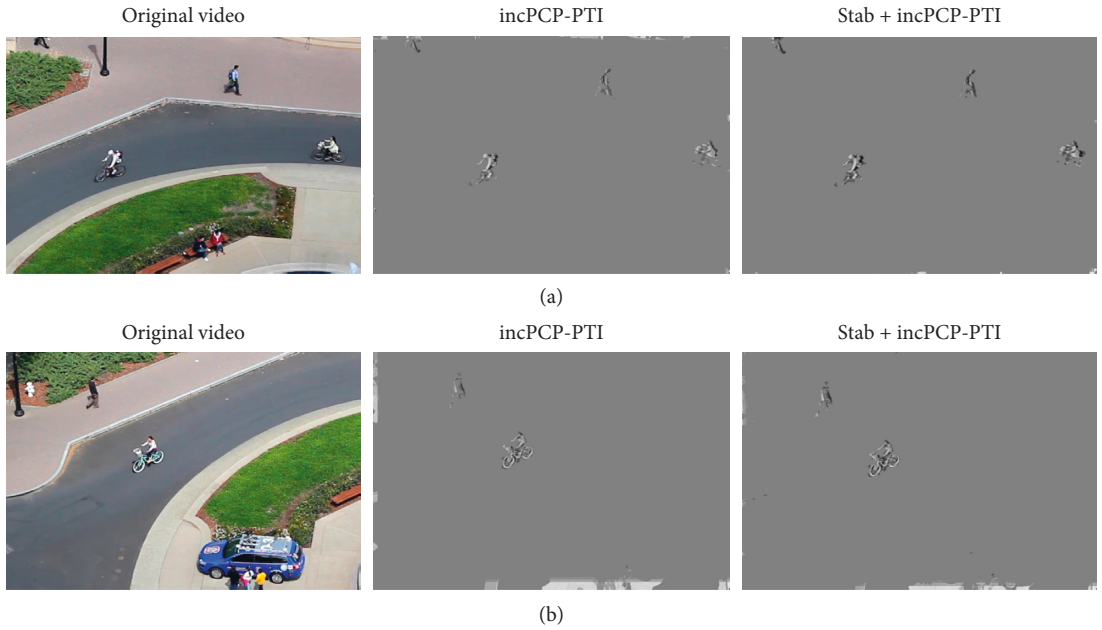

FIGURE 3: Representative frames of the video and the segmented sparse components for the SPJ dataset. Frames (a) 100 and (b) 355. This figure is a slightly modified version of Figure 3 in [28].

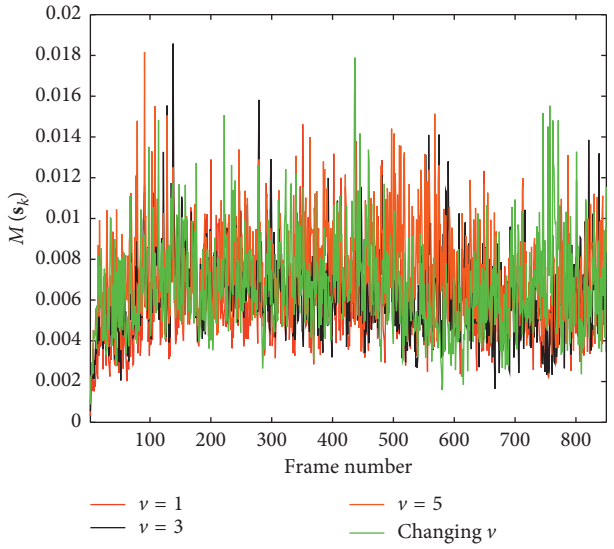

FIGURE 4: Value of distance $M(\mathbf{s}_k)$ between binary mask and ground truth frames for incPCP-PTI on the SPJ dataset (Figure 5). This figure is a slightly modified version of Figure 4 in [28].

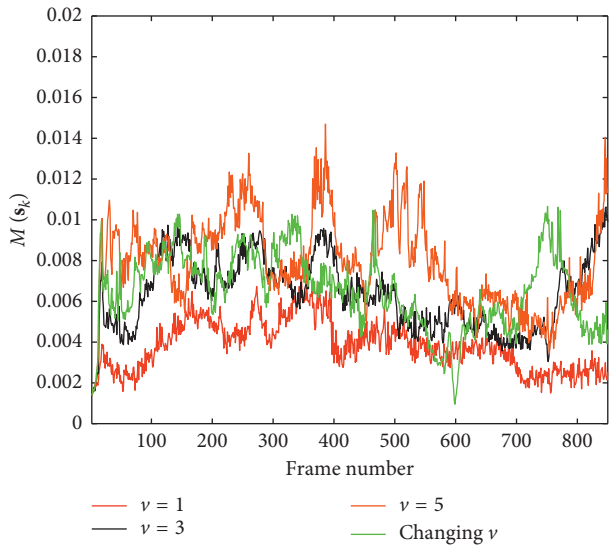

FIGURE 5: Value of distance $M(\mathbf{s}_k)$ between binary mask and ground truth frames for stab + incPCP-PTI on the SPJ dataset (Figure 4 and Section 4.1). This figure is a slightly modified version of Figure 5 of [28].

interactions. On the contrary, incPCP-PTI shows an average performance superior to LTVA-aPP and comparable to DECOLOR, though the latter did not run on all tested sequences.

The representative frames of Figures 8 and 9 exhibit different positions of the PTZ camera and thus evidence the ability of incPCP-PTI of handling the panning movements in the scene. IncPCP-PTI presents a relatively good F measure for both videos. This metric tended to be higher for the color version of the algorithm. In Figure 11, it can be observed that the F measure suffers decays at specific intervals of the video that coincide with sudden movements of the PTZ camera. However, after these sudden movements, the algorithm is able to restabilize and perform correctly. On the contrary, LTVA fails to track moving objects in a large number of frames. The lower performance on the CDNet dataset might be caused by the higher speed moving objects and panning movements that complicate the optical flow tracking using by LTVA. Additionally, the higher complexity of the objects and panning movements causes the clustering stage to produce a large number of false positives.

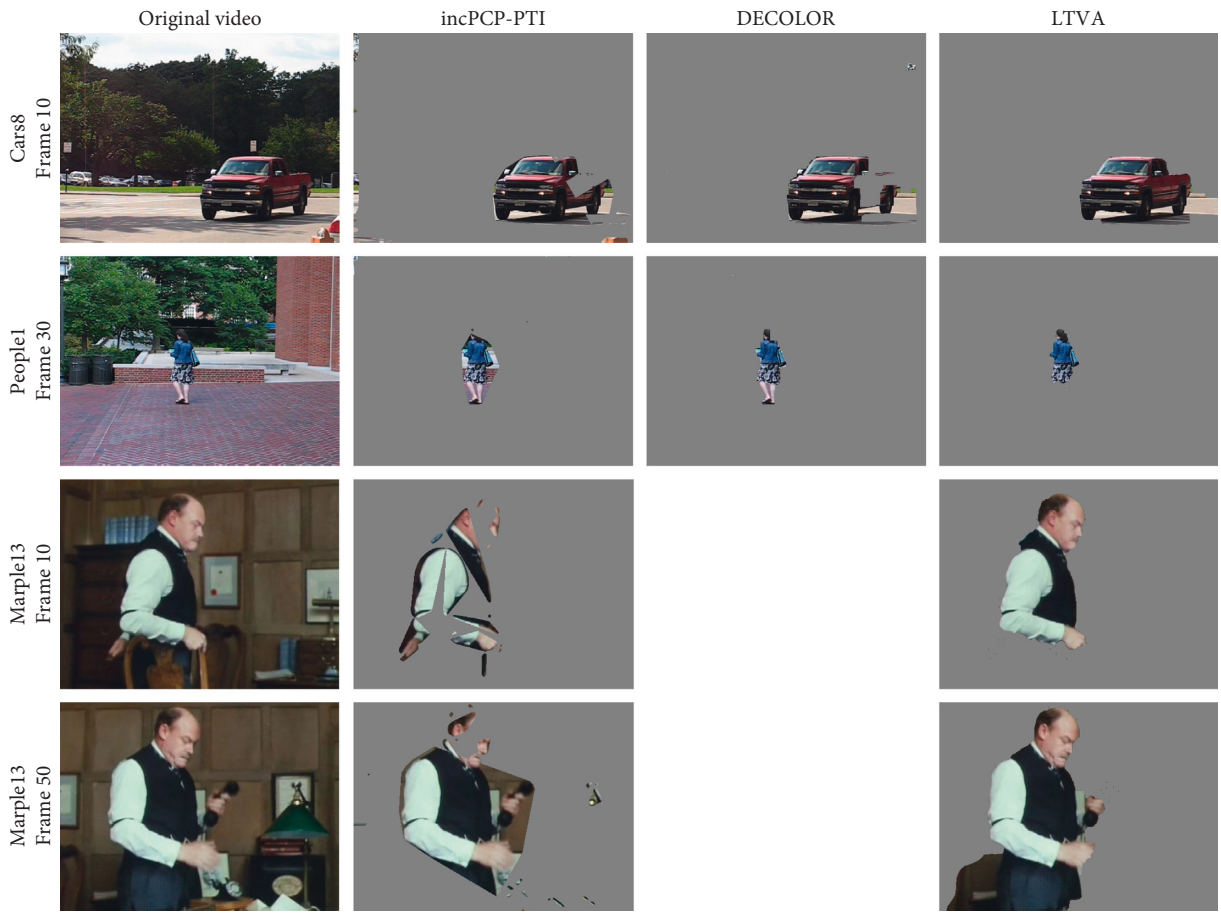

Figure 6: Selected frames from three sequences of the Moseg dataset (Section 4.2) along with the segmented sparse components for incPCP-PTI, DECOLOR, and LTVA. The sequences where DECOLOR did not work properly are left blank.

Table 2: Average F measure performance of incPCP-PTI, DECOLOR, and LTVA on the short panning Moseg sequences.

Sequence	IncPCP-PTI	DECOLOR	LTVA
cars1	0.78	0.50	0.90
cars2	0.42	0.66	0.75
cars3	0.76	0.83	0.93
cars4	0.77	0.81	0.92
cars5	0.73	0.82	0.86
cars6	0.73	0.75	0.94
cars7	0.72	0.84	0.92
cars8	0.74	0.47	0.85
cars9	0.50	0.41	0.49
people1	0.70	0.94	0.75
Average	0.68	0.70	0.83

Table 3: Average F measure performance of incPCP-PTI, DECOLOR, and LTVA on the long panning Moseg sequences.

Sequence	IncPCP-PTI	Decolor	LTVA
Marple1	0.29	—	0.87
Marple3	0.12	—	0.29
Marple7	0.22	—	0.37
Marple10	0.25	—	0.11
Marple13	0.55	—	0.82
Average	0.29	—	0.49

For both CP and IP videos (described in Section 4.4), incPCP-PTI showed a higher F measure than stab + incPCP-PTI, although a possible explanation is the misalignment of the ground truth reference frame and the reference frame of the stabilization algorithm. Nevertheless, the visual inspection of the frames and the results from the SPJ dataset suggests that incPCP-PTI is able to handle the presence of jitter in a panning scenario and that it does not need a stabilization preprocessing step. Compared to EFIC, incPCP-PTI showed superior performance in F measure in the CP videos, even without the postprocessing stage. In the IP video, incPCP-PTI is comparable or superior in F measure when compared with EFIC. As mentioned, the absence of open code for EFIC makes it difficult to make a more throughout comparison and to draw further conclusion from these comparisons. Compared to LTVA, incPCP-PTI shows a much higher F measure in both cases. These results suggest that incPCP-PTI might be more adequate than LTVA to track fastest panning and more complex scenarios. It is also noticed that incPCP-PTI attains these results in an incremental manner and with comparable or less computational average time per frame, despite the fact that the LTVA public code implemented in C and CUDA.

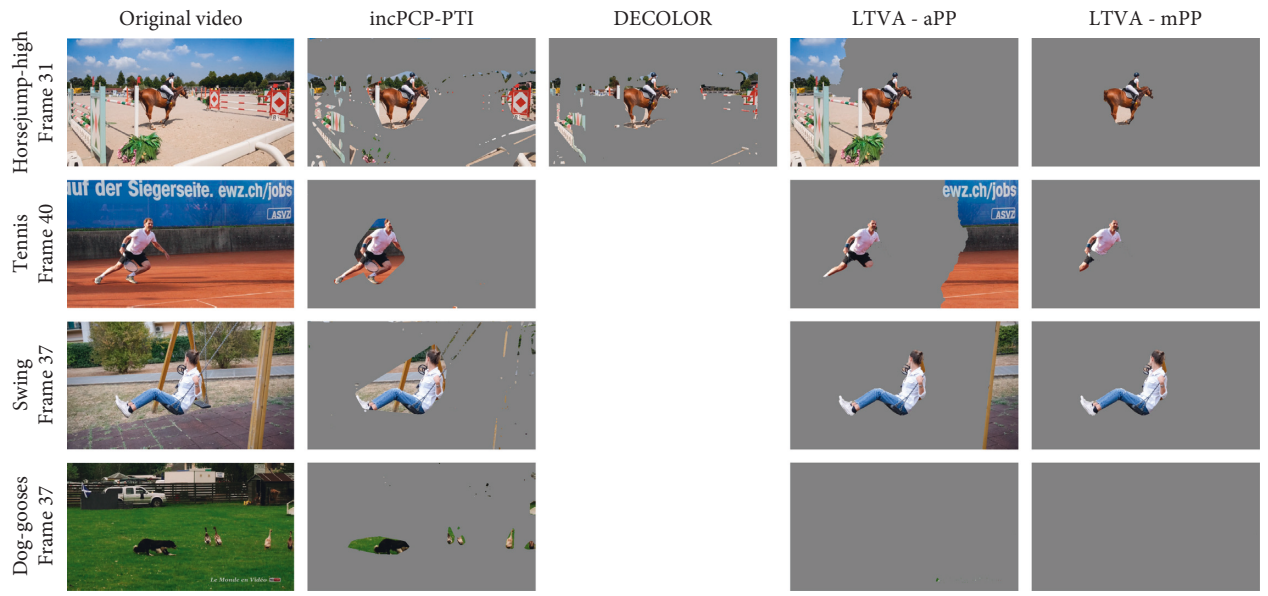

Figure 7: Selected frames from three sequences of the Davis dataset (Section 4.3) along with the segmented sparse components for incPCP-PTI, DECOLOR, and LTVA (aPP and mPP). The sequences in which DECOLOR did not work properly are left blank.

Table 4: Average F measure performance of incPCP-PTI, DECOLOR, and LTVA on the DAVIS dataset sequences.

	IncPCP-PTI	DECOLOR	LTVA-aPP	LTVA-mPP
Tennis	0.60	—	0.15	0.73
Soapbox	0.53	—	0.48	0.79
Bmx-bumps	0.31	—	0.30	0.46
Horsejump-high	0.46	0.50	0.22	0.82
Dance-jump	0.19	0.51	0.30	0.30
Swing	0.51	—	0.37	0.88
Dog-gooses	0.55	—	0.02	0.02
Skate-park	0.27	0.17	0.15	0.59
Bmx-trees	0.35	—	0.46	0.46
Scooter-gray	0.29	—	0.40	0.85
Average	0.40	0.39	0.28	0.59

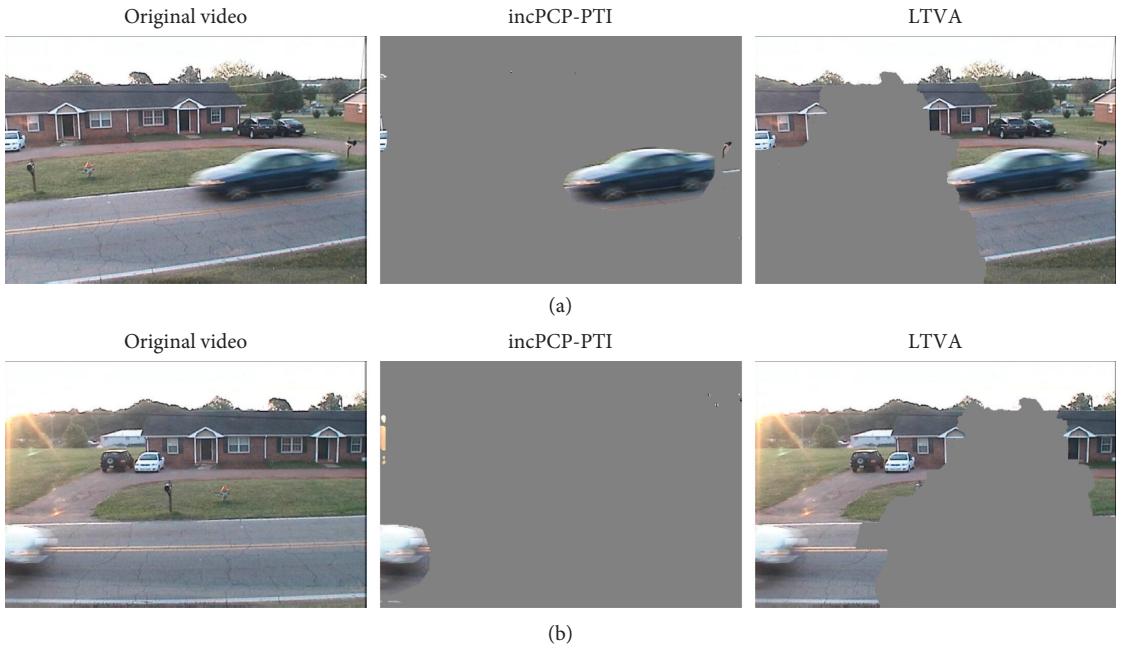

Figure 8: Frames 988 (a) and 1008 (b) of the video and the segmented sparse components for the CP video obtained with both incPCP-PTI and LTVA. It is observed that LTVA is unable to separate the moving objects any of the frames. This figure is a modified version of Figure 6 in [28].

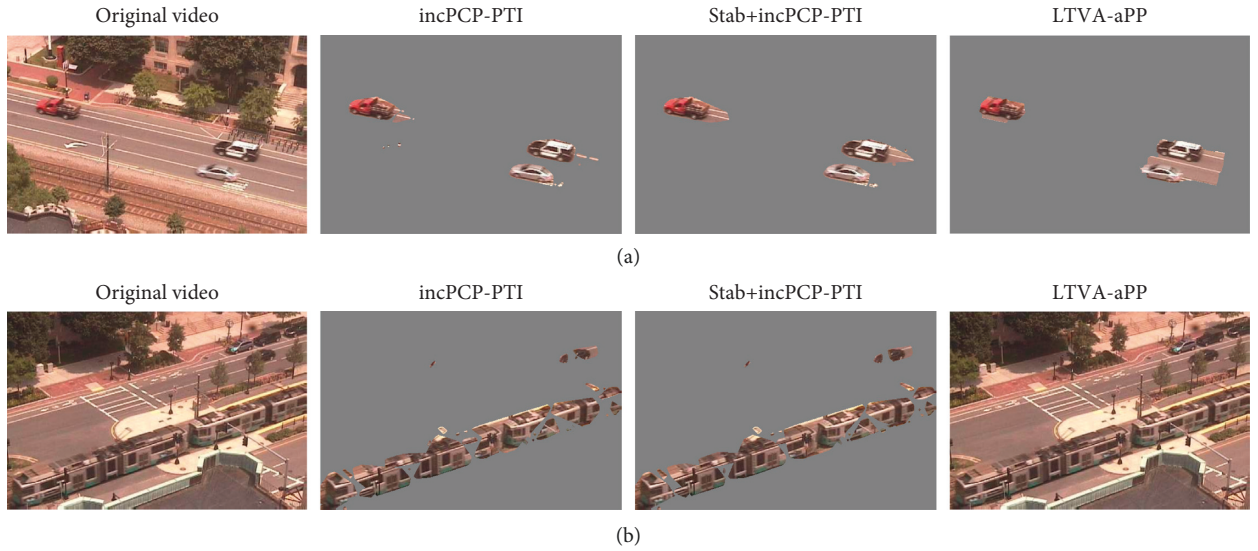

Figure 9: Frames 1330 (a) and 1870 (b) of the video and the segmented sparse components for the IP video obtained with incPCP-PTI, stab + incPCP-PTI, and LTVA-aPP. It is observed that LTVA is unable to separate the moving objects in frame 1870. This figure is a modified version of Figure 7 in [28].

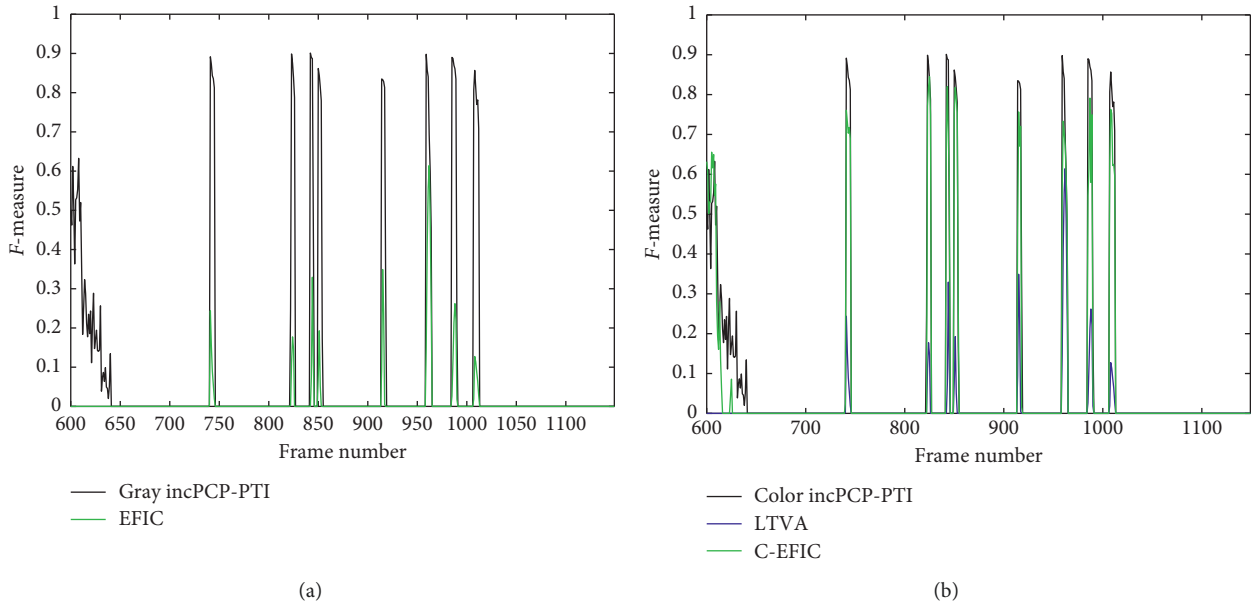

Figure 10: F measure per frame for the CP video for the grayscale-based methods (a) and color methods (b). Note: shown only for available frames (restriction of dataset). This figure is a slightly modified version of ([28], Figure 8).

7. Conclusion

We have presented a novel algorithm, incPCP-PTI, and have shown with artificial datasets and real videos from the Moseg, DAVIS, and CDnet2014 datasets that it can adequately detect moving objects in scenarios with simultaneous panning and jitter. To the best of our knowledge, this is the first incremental PCP-like method able to handle panning conditions. For the synthetic datasets, the algorithm maintained a low distance with respect to a proxy ground truth, and for the real videos, it maintained an adequate F measure and was able to stabilize after sudden panning of the camera. Additionally, the comparisons with stab + incPCP-PTI (independent video stabilization followed by incPCP-PTI) suggest that a stabilization stage preceding incPCP-PTI is not needed, as it is able to handle the jitter present in the camera motions. The evaluations on real videos show that the incPCP-PTI might be comparable or superior, depending on the case, to the state-of-the-art batch PCP (e.g., DECOLOR) and non-PCP-like (e.g., LTVA, EFIC) foreground separation methods.

Further improvements of the algorithm might focus on (i) making it able to handle other types of distortion-like perspective changes or zooming in/out of the camera and

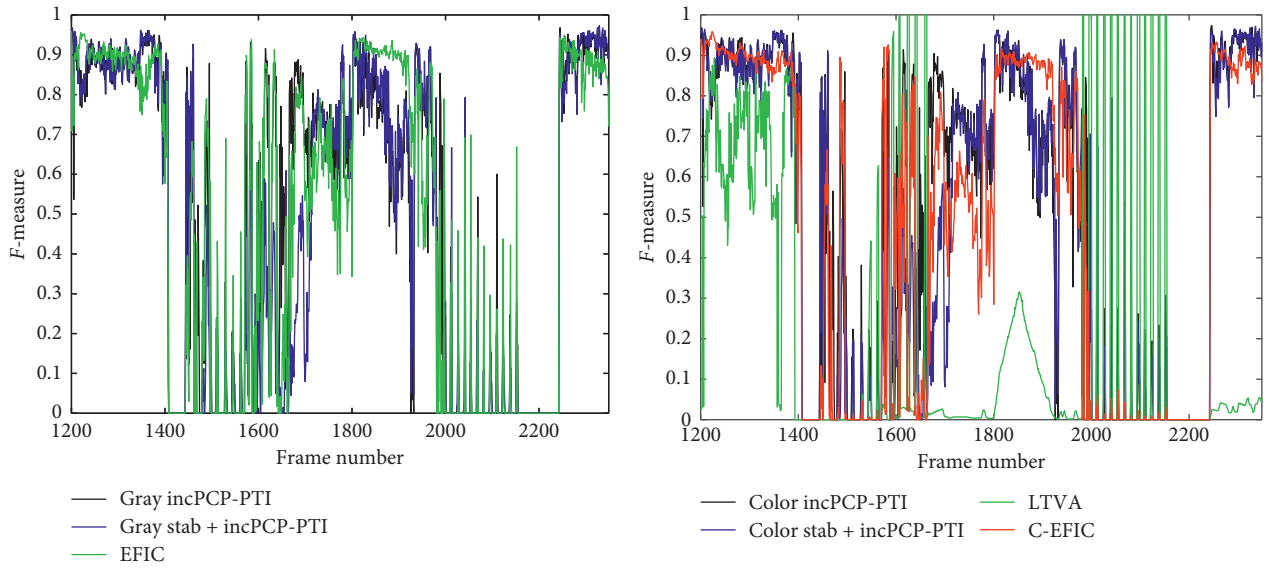

Figure 11: F measure per frame for the IP video for the grayscale-based methods (a) and color methods (b). Note: they are shown only for available frames (restriction of dataset). This figure is a slightly modified version of Figure 9 of [28].

Table 5: Value of F measure for grayscale and color incPCP-PTI and for EFIC and C-EFIC on the CP video.

Method	F measure	Average time per frame (seconds)
Grayscale incPCP-PTI	0.50	2.10
Color incPCP-PTI	0.49	3.58
LTVA-aPP	0.07	3.32
EFIC	0.42	—
C-EFIC	0.46	—

Table 6: Value of F measure for grayscale and color incPCP-PTI and stab + incPCP-PTI on the IP video.

Method	Average F measure	Average time per frame (seconds)
Grayscale incPCP-PTI	0.69	1.41
Color incPCP-PTI	0.70	2.31
Grayscale stab + incPCP-PTI	0.63	(89) + (1.41)
Color stab + incPCP-PTI	0.64	(89) + (2.31)
LTVA-aPP	0.27	9.01
EFIC	0.68	—
C-EFIC	0.64	—

Table 7: Value of average distance $\overline{M(\mathbf{s}_k)}$ for incPCP-PTI, stab + incPCP-PTI, and baseline incPCP (Section 4.1) on the SPJ dataset.

Dataset	incPCP-PTI	stab + incPCP-PTI	Baseline incPCP
$v = 1$	0.0057	0.0038	0.0015
$v = 3$	0.0064	0.0064	0.0021
$v = 5$	0.0071	0.0079	0.0022
Changing v	0.0066	0.0065	0.0024

(ii) reduce the time it takes per frame in order to make it more readily accessible for high frame rate real-time applications.

Data Availability

The video data used to support the findings of this study are included within the article. The datasets used in the article are referenced and can be found at publically available sites, namely, (1) synthetic datasets (Section 4.1) were constructed from: USC Neovision2 Project, https://goo.gl/5Si2Nm; (2) Moseg dataset (Section 4.2): "Freiburg-Berkeley motion segmentation dataset," https://goo.gl/bzEvvi; (3) DAVIS dataset (Section 4.3): "DAVIS: Densely Annotated VIdeo Segmentation," https://goo.gl/G8Hb7o; (4) CDnet2014 dataset (Section 4.4): http://www.changedetection.net/. Additionally, our implemented method can be found at http://goo.gl/4jEvck.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This research was supported by the "Programa Nacional de Innovación para la Competitividad y Productividad" (Innóvate Perú) Program, 169-Fondecyt-2015.

References

[1] Y. Xu, J. Dong, B. Zhang, and D. Xu, "Background modeling methods in video analysis: a review and comparative evaluation," *CAAI Transactions on Intelligence Technology*, vol. 1, no. 1, pp. 43–60, 2016.

[2] S. Calderara, R. Cucchiara, and A. Prati, "A distributed outdoor video surveillance system for detection of abnormal people trajectories," in *Proceedings of ACM/IEEE International Conference on Distributed Smart Cameras*, pp. 364–371, Vienna, Austria, September 2007.

[3] T. Bouwmans, F. Porikli, B. Höferlin, and A. Vacavant, *Background Modeling and Foreground Detection for Video Surveillance*, Chapman and Hall/CRC, Boca Raton, FL, USA, 2014.

[4] Z. Zivkovic, "Improved adaptive Gaussian mixture model for background subtraction," in *Proceedings of the 17th International Conference on Pattern Recognition (ICPR 2004)*, vol. 2, pp. 28–31, Cambridge, UK, August 2004.

[5] A. Elgammal, D. Harwood, and L. Davis, *Non-Parametric Model for Background Subtraction*, Springer Berlin Heidelberg, Berlin, Germany, 2000.

[6] L. Maddalena and A. Petrosino, "A self-organizing approach to background subtraction for visual surveillance applications," *IEEE Transactions on Image Processing*, vol. 17, no. 7, pp. 1168–1177, 2008.

[7] M. Shah, J. D. Deng, and B. J. Woodford, "Video background modeling: recent approaches, issues and our proposed techniques," *Machine Vision and Applications*, vol. 25, no. 5, pp. 1105–1119, 2013.

[8] T. Bouwmans and E. H. Zahzah, "Robust PCA via principal component pursuit: a review for a comparative evaluation in video surveillance," *Computer Vision and Image Understanding*, vol. 122, pp. 22–34, 2014.

[9] J. Wright, A. Ganesh, S. Rao, Y. Peng, and Y. Ma, "Robust principal component analysis: exact recovery of corrupted low-rank matrices via convex optimization," in *Proceedings of Advances in NIPS*, pp. 2080–2088, Vancouver, BC, Canada, December 2009.

[10] Z. Lin, M. Chen, and Y. Ma, "The augmented Lagrange multiplier method for exact recovery of corrupted low-rank matrices," 2010, http://arxiv.org/abs/1009.5055.

[11] G. Liu, Z. Lin, and Y. Yu, "Robust subspace segmentation by low-rank representation," in *Proceedings of 27th International Conference on Machine Learning (ICML 2010)*, pp. 663–670, Haifa, Israel, June 2010.

[12] T. Bouwmans, A. Sobral, S. Javed, S. K. Jung, and E.-H. Zahzah, "Decomposition into low-rank plus additive matrices for background/foreground separation: a review for a comparative evaluation with a large-scale dataset," *Computer Science Review*, vol. 23, pp. 1–71, 2017.

[13] P. Rodríguez and B. Wohlberg, "Fast principal component pursuit via alternating minimization," in *Proceedings of IEEE International Conference on Image Processing (ICIP 2013)*, pp. 69–73, Melbourne, VIC, Australia, September 2013.

[14] P. Rodriguez and B. Wohlberg, "Incremental principal component pursuit for video background modeling," *Journal of Mathematical Imaging and Vision*, vol. 55, no. 1, pp. 1–18, 2015.

[15] P. Rodriguez and B. Wohlberg, "A matlab implementation of a fast incremental principal component pursuit algorithm for video background modeling," in *Proceedings of IEEE International Conference on Image Processing (ICIP 2014)*, pp. 3414–3416, Paris, France, October 2014.

[16] J. He, L. Balzano, and A. Szlam, "Incremental gradient on the grassmannian for online foreground and background separation in subsampled video," in *Proceedings of IEEE CVPR*, pp. 1568–1575, Providence, RI, USA, 2012.

[17] J. Feng, H. Xu, and S. Yan, "Online robust PCA via stochastic optimization," in *Proceedings of Advances in NIPS*, pp. 404–412, Lake Tahoe, NV, USA, December 2013.

[18] N. Srebro, J. Rennie, and T. Jaakola, "Maximum-margin matrix factorization," in *Proceedings of Advances in NIPS*, pp. 1329–1336, MIT Press, Vancouver, BC, Canada, December 2005.

[19] M. Rahmani and G. Atia, "High dimensional low rank plus sparse matrix decomposition," *IEEE Transactions on Signal Processing*, vol. 65, no. 8, pp. 2004–2019, 2017.

[20] M. Yazdi and T. Bouwmans, "New trends on moving object detection in video images captured by a moving camera: a survey," *Computer Science Review*, vol. 28, pp. 157–177, 2018.

[21] Y. Peng, A. Ganesh, J. Wright, W. Xu, and Y. Ma, "RASL: robust alignment by sparse and low-rank decomposition for linearly correlated images," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 11, pp. 2233–2246, 2012.

[22] J. He, D. Zhang, L. Balzano, and T. Tao, "Iterative grassmannian optimization for robust image alignment," *Image and Vision Computing*, vol. 32, no. 10, pp. 800–813, 2014.

[23] P. Rodríguez and B. Wohlberg, "Translational and rotational jitter invariant incremental principal component pursuit for video background modeling," in *Proceedings of IEEE International Conference on Image Processing (ICIP 2015)*, pp. 537–541, Quebec City, QC, Canada, September 2015.

[24] C. Chen, S. Li, H. Qin, and A. Hao, "Robust salient motion detection in non-stationary videos via novel integrated strategies of spatio-temporal coherency clues and low-rank analysis," *Pattern Recognition*, vol. 52, pp. 410–432, 2016.

[25] X. Zhou, C. Yang, and W. Yu, "Moving object detection by detecting contiguous outliers in the low-rank representation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 3, pp. 597–610, 2013.

[26] S. Ebadi, V. Ones, and E. Izquierdo, "Efficient background subtraction with low-rank and sparse matrix decomposition," in *Proceedings of IEEE IEEE International Conference on Image Processing (ICIP 2015)*, pp. 4863–4867, IEEE, Quebec City, QC, Canada, September 2015.

[27] C. Gao, B. E. Moore, and R. R. Nadakuditi, "Augmented robust pca for foreground-background separation on noisy, moving camera video," 2017, http://arxiv.org/abs/1709.09328.

[28] G. Chau and P. Rodrıguez, "Panning and jitter invariant incremental principal component pursuit for video background modeling," in *Proceedings of IEEE International Conference on Computer Vision (ICCV 2017)*, pp. 1844–1852, Venice, Italy, October 2017.

[29] P. Ochs, J. Malik, and T. Brox, "Segmentation of moving objects by long term video analysis," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, no. 6, pp. 1187–1200, 2014.

[30] J. Pont-Tuset, S. Caelles, F. Perazzi et al., "The 2018 davis challenge on video object segmentation," 2018, http://arxiv.org/abs/1803.00557.

[31] Y. Wang, P. Jodoin, F. Porikli, J. Konrad, Y. Benezeth, and P. Ishwar, "CDnet 2014: an expanded change detection benchmark dataset," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPR 2014)*, pp. 387–394, Columbus, OH, USA, June 2014.

[32] S. Geman and D. Geman, "Stochastic relaxation, gibbs distributions, and the bayesian restoration of images," in *Readings in Computer Vision*, pp. 564–584, Elsevier, Amsterdam, Netherlands, 1987.

[33] J.-M. Odobez and P. Bouthemy, "Robust multiresolution estimation of parametric motion models," *Journal of Visual Communication and Image Representation*, vol. 6, no. 4, pp. 348–365, 1995.
[34] J. Bunch and C. Nielsen, "Updating the singular value decomposition," *Numerische Mathematik*, vol. 31, no. 2, pp. 111–129, 1978.
[35] Y. Chahlaoui, K. Gallivan, and P. Van Dooren, "Computational information retrieval," in *Computational Information Retrieval*, An Incremental Method for Computing Dominant Singular Spaces, pp. 53–62, SIAM, India, 2001.
[36] C. Baker, K. Gallivan, and P. V. Dooren, "Low-rank incremental methods for computing dominant singular subspaces," *Linear Algebra and Its Applications*, vol. 436, no. 8, pp. 2866–2888, 2012.
[37] M. Brand, "Fast low-rank modifications of the thin singular value decomposition," *Linear Algebra and its Applications*, vol. 415, no. 1, pp. 20–30, 2006.
[38] D. Goldfarb and S. Ma, "Convergence of fixed-point continuation algorithms for matrix rank minimization," *Foundations of Computational Mathematics*, vol. 11, no. 2, pp. 183–210, 2011.
[39] E. Candès, X. Li, Y. Ma, and J. Wright, "Robust principal component analysis?," *Journal of the ACM*, vol. 58, no. 3, pp. 1–37, 2011.
[40] P. Rodrıguez and B. Wohlberg, "An incremental principal component pursuit algorithm via projections onto the l1 ball," in *Proceedings of XXIV International Conference on Electronics, Electrical Engineering and Computing (INTERCON)*, Cusco, Peru, August 2017.
[41] J. Duchi, S. Shalev-Shwartz, Y. Singer, and T. Chandra, "Efficient projections onto the l1-ball for learning in high dimensions," in *Proceedings of 25th International Conference on Machine Learning (ICML)*, pp. 272–279, New York, NY, USA, 2008.
[42] L. Condat, "Fast projection onto the simplex and the ℓ_1 ball," *Mathematical Programming*, vol. 158, no. 1, pp. 575–585, 2016.
[43] P. Rodriguez, "Accelerated gradient descent method for projections onto the l1-ball," in *Proceedings of IEEE 13th Image, Video, and Multidimensional Signal Processing Workshop (IVMSP)*, Zagorochoria, Greece, June 2018.
[44] P. Rodriguez, "An accelerated Newton's method for projections onto the l1-ball," in *Proceedings of 2017 IEEE 27th International Workshop on Machine Learning for Signal Processing (MLSP)*, pp. 1–6, Tokyo, Japan, September 2017.
[45] P. Rodríguez and B. Wohlberg, "Ghosting suppression for incremental principal component pursuit algorithms," in *Proceedings of IEEE GlobalSIP*, pp. 197–201, Washington, DC, USA, December 2016.
[46] DAVIS: densely annotated VIdeo segmentation," https://goo.gl/G8Hb7o.
[47] J. Dong and H. Liu, "Video stabilization for strict real-time applications," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 27, no. 4, pp. 716–724, 2017.
[48] USC Neovision2 Project, http://goo.gl/5Si2Nm.
[49] Freiburg-berkeley motion segmentation dataset," https://goo.gl/bzEvvi.
[50] T. Brox and J. Malik, "Object segmentation by long term analysis of point trajectories," in *Proceedings of European Conference on Computer Vision*, pp. 282–295, Springer, Crete, Greece, September 2010.
[51] J. Chassery and C. Garbay, "An iterative segmentation method based on a contextual color and shape criterion," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 6, no. 6, pp. 794–800, 1984.
[52] Object segmentation by long term analysis of point trajectories," https://goo.gl/VtdQbb.
[53] Detecting contiguous outliers in the low-rank representation," https://goo.gl/xAp1Nc.
[54] G. Allebosch, F. Deboeverie, P. Veelaert, and W. Philips, "EFIC: edge based foreground background segmentation and interior classification for dynamic camera viewpoints," in *Proceedings of 16th International Conference on Advanced Concepts for Intelligent Vision Systems (ACIVS)*, Springer, Catania, Italy, October 2015.
[55] G. Allebosch, D. Van Hamme, F. Deboeverie, P. Veelaert, and W. Philips, "C-EFIC: color and edge based foreground background segmentation with interior classification," in *Proceedings of VISIGRAPP*, pp. 433–454, Springer, Berlin, Germany, March 2015.
[56] Results for CDnet 2014, http://goo.gl/SSBvFA.
[57] EFIC results, http://goo.gl/LQBeKR.
[58] C-EFIC results, http://goo.gl/ctqmNs.