This ICCV workshop paper is the Open Access version, provided by the Computer Vision Foundation.
Except for this watermark, it is identical to the version available on IEEE Xplore.

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

Gustavo Chau
Department of Electrical Engineering
Pontificia Universidad Catolica del Peru
gustavo.chau@pucp.edu.pe

Paul Rodríguez
Department of Electrical Engineering
Pontificia Universidad Catolica del Peru
prodrig@pucp.edu.pe

## Abstract

*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. We present a novel, fully incremental PCP algorithm, named incPCP-PTI, that is 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. Results on synthetic videos and CDNET2014 videos 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.*

## 1. Introduction

Video background modeling consists on segmenting the "foreground" or moving objects from the static "background". It is an important first step in various computer vision applications [39] such as abnormal event identification [7] and surveillance [3].

Several video background modeling methods using different approaches such as Gaussian mixture models [41], kernel density estimations [16] or neural networks [23] exist in the literature. More comprehensive surveys of other methods are presented in [34] and [39]. Principal Component Pursuit (PCP) is currently considered to be one of the leading algorithms for video background modeling [5]. Formally, PCP was introduced in [38] as the non-convex optimization problem

$$\underset{L,S}{\operatorname{argmin}} \quad \operatorname{rank}(L) + \lambda\|S\|_0 \quad \text{s.t.} \quad D = L + S, \tag{1}$$

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, $\lambda$ is a fixed global regularization parameter, and $\operatorname{rank}(L)$ is the rank of $L$ and $\|S\|_0$ is the $\ell_0$ norm of $S$.

Although the convex relaxation given in (2)

$$\underset{L,S}{\operatorname{argmin}} \|L\|_* + \lambda\|S\|_1 \ \text{ s.t. } D = L + S\ , \tag{2}$$

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 $\ell_1$ norm of $S$, is at the core of most PCP algorithms (including the Augmented Lagrange Multiplier (ALM) and inexact ALM (iALM) algorithms, see [21, 22]), there exists several others (for a complete list, see [4, Table 4]). In particular, we point out

$$\underset{L,S}{\operatorname{argmin}} \frac{1}{2}\|L+S-D\|_F^2 + \lambda\|S\|_1 \quad \text{s.t. } \operatorname{rank}(L) \le r, \tag{3}$$

where $\|\cdot\|_F^2$ is the Frobenius norm, which was originally proposed in [28], since we will use it as the starting point of our proposed method (see Sections 2.2 and 3.1).

In [5] it was showed that PCP provides state-of-the-art performance in video background modeling problems, but also some of its limitations were stated.

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 [32, 29] (applied to (3)), by low-rank subspace tracking [19] (applied to (2)), stochastic optimization [17] (which applies the Maximum-Margin Matrix Factorization (M3F) method, [35], to (2)), or random sampling [25] (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. The Robust Alignment by Sparse and Low-Rank decomposition (RASL) method [24], which used (2) as its starting point, 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 other hand, t-GRASTA [20] and incPCP-TI [30], 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 [10, 40, 14], but all of them are batch or semi-batch methods; furthermore, all of them used (2) as their starting point and also used the same general ideas (see (4)) as RASL. Therefore, a fully on-line 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 propose to address the panning problem by modifying the optimization problem solved by incPCP-TI [32], which in turn uses (3) as its starting point, and applying a set of transformations to the low-rank component that are updated with each incoming new frame. Our computational experiments on synthetically created datasets and publicly available videos of the CDNET2014 [37] dataset 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

In this section, two previous on-line or partially on-line 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.

### 2.1. t-GRASTA

The Grassmannian Robust Adaptive Subspace Tracking Algorithm (GRASTA) [19] is a semi on-line method for low-rank subspace tracking that has been applied to the foreground-background separation problem. GRASTA is not a fully on-line 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 [20] and it is based on the Robust Alignment by Sparse and Low-Rank decomposition (RASL) algorithm [24]. RASL tries to handle the misalignment in the video frames by solving

$$\underset{L,S,\tau}{\operatorname{argmin}} \|L\|_* + \lambda\|S\|_1 \quad s.t. \quad \tau(D) = L + S, \tag{4}$$

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 to (2).

The non-linearity in the transformations $\tau$ of (4) are 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 $\tau$ 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. incPCP-TI

The incPCP-TI [30] considers the optimization problem

$$\begin{aligned} \underset{L^*,S,\mathcal{T}}{\operatorname{argmin}} \quad & \tfrac{1}{2}\|D - \mathcal{T}(L^*) - S\|_2^2 + \lambda\|S\|_1 \\ \text{s.t.} \quad & \operatorname{rank}(L^*) \leq r, \end{aligned} \tag{5}$$

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, i.e.:

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

where $D^*$ represents the un-observed jitter-free video sequence, $H = \{h_k\}$ is a set filters that independently models translation for each frame, $*$ represents convolutiona 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 $\mathcal{T} = \tau^{-1}$, i.e. the transformation used in (5) can be understood as the inverse of the transformation used in RASL or t-GRATSA (see (4)).

In [32, 29] a computationally efficient and fully incremental algorithm, based on rank-1 updates for thin SVD, was proposed to solve (3); in [30] 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

$$\begin{aligned} \underset{L,S,H,\alpha}{\operatorname{argmin}} \quad & \frac{1}{2}\sum_k \|h_k * R(\mathbf{l}_k^*, \alpha_k) + \mathbf{s}_k - \mathbf{d}_k\|_F^2 + \lambda\|S\|_1 \\ & +\gamma\sum_k \|h_k\|_1 \quad \text{s.t. } \operatorname{rank}(L^*) \leq r; \end{aligned} \tag{7}$$

can be efficiently computed in an incremental fashion (see [30, Section 3.3] for details).

## 3. Methods

### 3.1. Proposed incPCP-PTI Method

The proposed algorithm (named incPCP-PTI) is a modification of the previously proposed incPCP-TI [30] so that it is able to handle panning and camera motion. 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 $\xi_k$ for the k-th frame. However, instead of using iterative hard thresholding (IHT) [18] as in the original incPCP-TI, the low-rank approximation problem is solved in the reference frame of $\mathbf{d}_k$ by applying $\xi_k^{-1}$ to the residual $\mathbf{d}_k - \mathbf{s}_k$. The whole procedure is presented in Algorithm 1. This algorithm makes use of the repSVD and downSVD operators, which correspond to the thin SVD replacement and downdate operators, respectively [32].

In line 3 of Algorithm 1, the latest low rank frame $l_k$ is aligned to the current frame $d_k$. The transformation is estimated as the composition of a translation and rotation. Such found align 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 $\xi_k$, due to jitter remain (line 10 of Algorithm 1).

The ghosting suppression mentioned in line 16 is detailed in subsection 3.3. The shrinkage in line 9 of Algorithm 1 can be performed by either soft-thresholding or projection on the $\ell_1$-ball. Soft-thresholding is performed with a simple element-wise shrinkage operator ($\text{shrink}(x, \lambda) = \text{sign}(x)\max(0, |x| - \lambda)$). Projection onto the $\ell_1$ ball is detailed in subsection 3.2. For all our experiments, the latter was chosen.

### 3.2. Projection on the $\ell_1$ ball

Although theoretical guidance is available for selecting a minimax optimal regularization parameter $\lambda$ in (2) [8], practical problems do not fully satisfy the idealized assumptions, and thus $\lambda$ often has to be heuristically tuned. This problem is also observed if (3) is used instead of (2).

To tackle this problem, [33] introduced the alternative convex relaxation of (1) given by

$$\begin{aligned} \underset{L,S}{\text{argmin}} \quad & \tfrac{1}{2}\|L + S - D\|_F^2 \\ \text{s.t.} \quad & \|S\|_1 \leq \mu,\ \text{rank}(L) \leq r, \end{aligned} \tag{8}$$

**Algorithm 1:** incpcp-PTI

**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 $\quad ++m$;
3 $\quad$ find $\mathcal{T}_k$ such that $||\mathcal{T}_k(\mathbf{l}_{k-1}) - \mathbf{d}_k||_2$ is minimized
4 $\quad$ obtain $L^* = \mathcal{T}_k(L) = [\mathcal{T}_k(\mathbf{l}_1), \ldots, \mathcal{T}_k(\mathbf{l}_{k-1})]$
5 $\quad [U_k, \Sigma_k, V_k] = \text{partialSVD}(L^*, r)$
6 $\quad [U_k, \Sigma_k, V_k] = \text{incSVD}\,(\mathbf{d}_k, U_k, \Sigma_k, V_k)$
7 $\quad$ **for** $j = 1 : iL$ **do**
8 $\quad\quad \mathbf{l}_k^* = U_k(:, 1 : r) * \Sigma_k * (V_k(end, :)')$
9 $\quad\quad \mathbf{s}_k = \text{shrink}(\mathbf{d}_k - \mathbf{l}_k^*)$
10 $\quad\quad \xi_k = \arg\min_\xi \ ||\xi(\mathbf{l}_k^*) - (\mathbf{d}_k - \mathbf{s}_k)||_2$
11 $\quad\quad$ **if** *j==iL* **then**
12 $\quad\quad\quad$ break
13 $\quad\quad \rho = \xi_k^{-1}(\mathbf{d}_k - \mathbf{s}_k)$
14 $\quad\quad [U_k, \Sigma_k, V_k] = \text{repSVD}\,(\mathbf{d}_k, \rho, U_k, \Sigma_k, V_k)$
15 $\quad$ **end**
16 $\quad$ Apply ghosting suppression
17 $\quad$ **if** $m \geq bL$ **then**
18 $\quad\quad \text{downSVD}\,(1\text{stcolumn}, U_k, \Sigma_k, V_k)$
19 $\quad$ Update $k$ if necessary
20 **end**

which can also be incrementally solved via rank-1 updates for thin SVD (as is the case of the incPCP and related algorithms [32, 29, 30]), however (8) has the advantage that a simple heuristic can be derived for the adaptive selection of $\mu$ for each frame. Furthermore, $\mu$ 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 $\mathbf{s}_k = \text{proj}_{\|\cdot\|_1}(\mathbf{d}_k - \mathbf{l}_k, \mu)$, where

$$\text{proj}_{\|\cdot\|_1}(\mathbf{u}, \mu) \triangleq \min_{\mathbf{x}} \frac{1}{2}\|\mathbf{x} - \mathbf{u}\|_2^2 \ \text{ s.t. } \|\mathbf{x}\|_1 \leq \mu. \tag{9}$$

Thus, for the shrinkage step, the solution is given by projections into the $\ell_1$-ball of radius $\mu$.

While there are several well-known and efficient algorithms that solve (9), such [13, 11], [33] used [27], a recently published algorithm for (9) that has a better computational performance than either [13] or [11].

Furthermore, [33] also proposed a simple schema for adapting $\mu_k$ with every frame, which is given by

$$\mu_k = \alpha \cdot \|\mathbf{d}_k - \mathbf{l}_k\|_1, \tag{10}$$

where $\alpha$ is a value between 0.5 and 0.75.

### 3.3. Ghosting suppression

Ghosting refers to when the foreground estimates includes phantoms or smear replicas from actual moving objects. In [31], they proposed a procedure for ghosting suppression in the incPCP algorithm which consists on 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 $\hat{\mathbf{d}}_k^{(n)} = \mathbf{d}_k \odot \mathbf{b}_k + \mathbf{l}_k \odot (1 - \mathbf{b}_k)$, where $\odot$ represents an Hadamard product, that is used to update the low-rank component. Additionally, if the procedure with the $\ell_1$ ball projection described in 3.2 is used for the shrinkage step, $\mu_k$ can be spatially adapted in order to reduce ghosting [33]. Based on the difference between the 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:

$$\mathbf{s}_k = (1 - \mathbf{m}_k) \odot \hat{\mathbf{s}}_k + \mathbf{m}_k \odot \tilde{\mathbf{s}}_k, \tag{11}$$

where $\hat{\mathbf{s}}_k = \text{proj}_{\|\cdot\|_1}(\mathbf{d}_k - \mathbf{l}_k, \mu_k)$ and $\tilde{\mathbf{s}}_k = \text{proj}_{\|\cdot\|_1}(\mathbf{m}_k \odot \hat{\mathbf{s}}_k, \mu_k^{(g)})$ and $\mu_k^{(g)} = \beta \cdot \|\mathbf{m}_k \odot \hat{\mathbf{s}}_k\|_1$, where $\text{prox}_{\|\cdot\|_1}(\cdot)$ is defined in (9); $\beta$ is suggested to take values between 0.1 and 0.3

## 4. Dataset and computational experiments description

For the evaluation of the proposed incPCP-PTI algorithm, two datasets were considered. The first dataset consisted on synthetic jitter and panning videos and the second one consisted on videos of real panning taken from the CDNet 2014 dataset [37]. Both 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, 8MB Cache, 32GB RAM) with a 12GB NVIDIA Tesla K40C GPU card.

### 4.1. Synthetic datasets

A dataset with synthetic panning and jitter was generated from the 3rd Tower video of the USC Neovision2 dataset [36], which consists on 900 frames of size 1920 $\times$ 1088 pixel at 25 fps. For this purpose, a subregion of 720 $\times$ 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 :

$$u[n] = \begin{cases} u[n-1] + v \cdot (1,1) & u[n-1]_x < Q \\ u[n-1] + v \cdot (1,0) & u[n-1]_x \geq Q \end{cases} \tag{12}$$

where $u[0]$ is the initial point (in this case 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. The panning velocity $v$ was taken as 1, 3, 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 the SP and SPJ datasets, the sparse approximation via the batch iALM method [21] 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 [5, Tables 6 and 7], this result is considered to be reliable. For these synthetic datasets, the performance of the proposed algorithm was measured in terms of the normalized $\ell_1$ distance

$$M(\mathbf{s}_k) = \frac{||\mathbf{s}_k^{gt} - \mathbf{s}_k||_1}{N}, \tag{13}$$

where $\mathbf{s}_k^{gt}$ and $\mathbf{s}_k$ are the groundtruth 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 groundtruth) to 1.

### 4.2. Real panning and jitter dataset

Two videos from the PTZ category of the CDNet 2014 dataset [37] were chosen:

- continuousPan(CP): 704 x 480 pixel, 1700 frames-color video containing a continuous panning of a PTZ camera. The video is almost jitter free.
- intermittentPan(IP): 560 x 368 pixel, 3500 frame-color of a PTZ camera that changes between two fixed positions. The video contains intermittent panning and additional real jitter.

For both the CP and IP videos, the binary mask obtained from the incPCP-PTI algorithm was compared to the

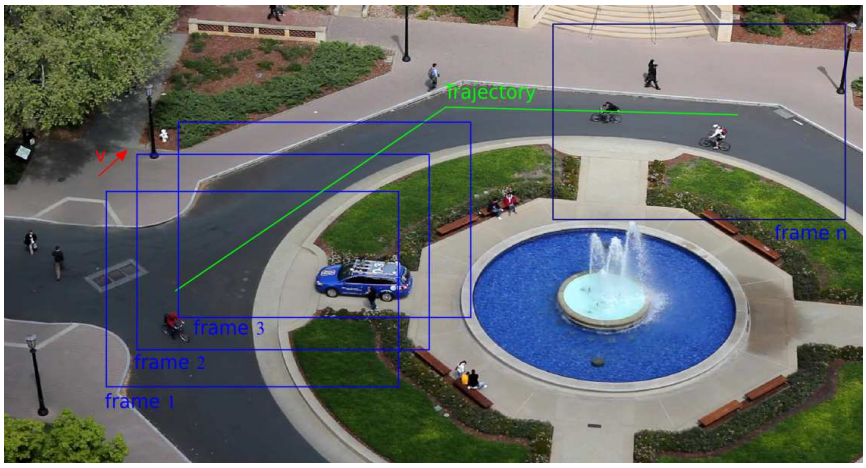


Figure 1. Construction of the synthetic panning and jitter dataset. The selected region (blue rectangles) were of 720 × 480 pixel and the centroid of the region was translated at a velocity $v$ (red vector) along a piecewise linear trajectory (green).

ground truth provided in the CDNet dataset in order to obtain an F-measure, defined as

$$F = \frac{2 \cdot P \cdot R}{P + R}, \; P = \frac{TP}{TP + FN}, \; R = \frac{TP}{TP + FP}, \quad (14)$$

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. The F-measure was evaluated only on frames that contained groudtruth motion.

### 4.3. Comparisons

To the best of our knowledge, no other 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. Accordingly, for the SP dataset only the incPCP-PTI method was evaluated. For the SPJ dataset, we evaluated incPCP-PTI and a method consisting on a preprocessing stage using a recent state-of-the-art video stabilization technique [12][1] followed by incPCP-PTI (this method will be denoted as stab+incPCP-PTI). This comparison had as objective determining if jitter is handled correctly by incPCP-PTI alone. Additionally, we include a baseline comparison with the sparse components obtained with incPCP on the full Neovision Tower video and then segmented used the same procedure described in subsection 4.1.

For the real videos, we also included a comparison with the Edge Based Foreground Background Segmentation and Interior Classification (EFIC) [1] and its color version, C-EFIC [2]. These methods were chosen as they obtained the second and third best F-measure in the PTZ category of the CDNET2014 dataset results [26]. 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 [15, 6]. 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, EFIC and C-EFIC include a post-processing step on the binary mask. Therefore, it was considered appropriate to also compare incPCP-PTI and stab+incPCP-PTI with a simple post-processing of the binary mask that corresponds to the computation of the convex hull of the connected objects [9]. For completeness, this post-processing after EFIC is also reported but, as noted, this method already entails a post-processing technique. For all incPCP-PTI variants, three inner loops and a window size of 30 background frames. For the $\ell_1$ ball projection, $\alpha$ was set to 0.75 and the ghosting suppression $n_2 - n_1$ was set to 20 frames. $\alpha$ controls the adaptation of $\tau$, with lower $\alpha$ forcing a sparser solution, whereas the difference $n_2 - n_1$ controls the number of frames used for ghosting suppression.

## 5. Results

### 5.1. Synthetic datasets

#### 5.1.1 SP dataset

The distance $M(\mathbf{s}_k)$ (see (13)) computed for each frame of the different videos of the SP dataset are 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 4.3. It can be noticed that the distance tends to increase as the panning velocity increases but the

[1]The authors of the paper provide a binary executable version of the algorithm in real-timedvs.blogspot.pe

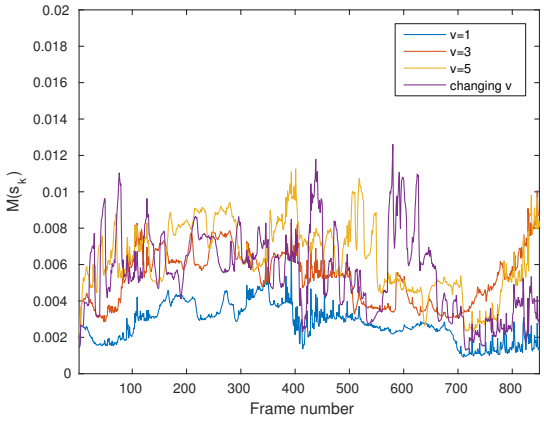


Figure 2. Value of distance $\delta$ between binary mask and groundtruth frames for incPCP-PTI on the SP dataset

Table 1. Value of average distance $\overline{M(\mathbf{s}_k)}$ between binary mask and groundtruth frames for incPCP-PTI and baseline incPCP (see Section 4.3) on the SP dataset. 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 |

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 Figure 4 and Figure 5 for incPCP-PTI and stab+incPCP-PTI, respectively. Table 2 exhibits the average distance $\overline{M(\mathbf{s}_k)}$ for the SPJ dataset for both methods along with a baseline metric, described in Section 4.3. In general, the distance for these videos is higher than the distance obtained for their non-jitter counterparts.

### 5.2. Real panning and jitter dataset

Representative frames of the video and the segmented sparse components for the CP and IP video are shown in Figures 6 and 7, respectively. Figure 8 shows the F-measure (with no post-processing) for incPCP-PTI (grayscale and color versions) and EFIC and C-EFIC on the frames of the CP video, while Figure 9 shows the same metric for all methods on the frames of the IP video. Tables 3 and 4 show

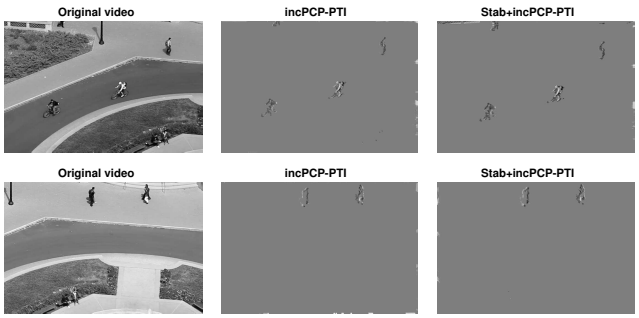


Figure 3. Representative frames of the video and the segmented sparse components for the SPJ dataset. Frame 100 is shown in the top row and frame 355 is shown in the bottom row

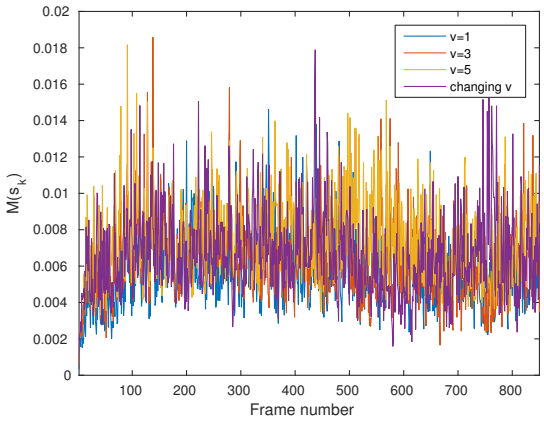


Figure 4. Value of distance $M(\mathbf{s}_k)$ between binary mask and groundtruth frames for incPCP-PTI on the SPJ dataset

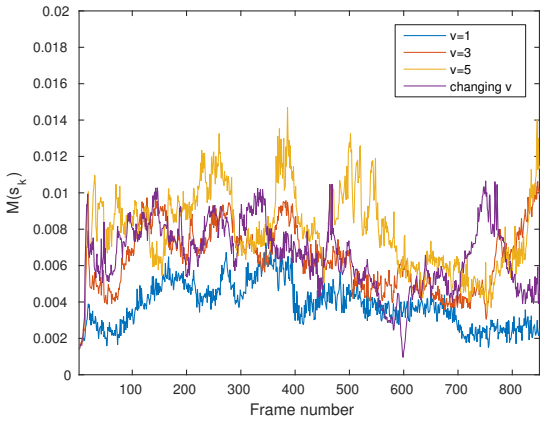


Figure 5. Value of distance $M(\mathbf{s}_k)$ between binary mask and groundtruth frames for stab+incPCP-PTI on the SPJ dataset

the average F-measure and computational time obtained over all frames. The F-measures with post-processing are shown in brackets. For stab+incPCP-PTI the computational time is shown as (Total stabilization time) + (incPCP-PTI

Table 2. Value of average distance $\overline{M(\mathbf{s}_k)}$ for incPCP-PTI, stab+incPCP-PTI and baseline incPCP (see Section 4.3) 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 |

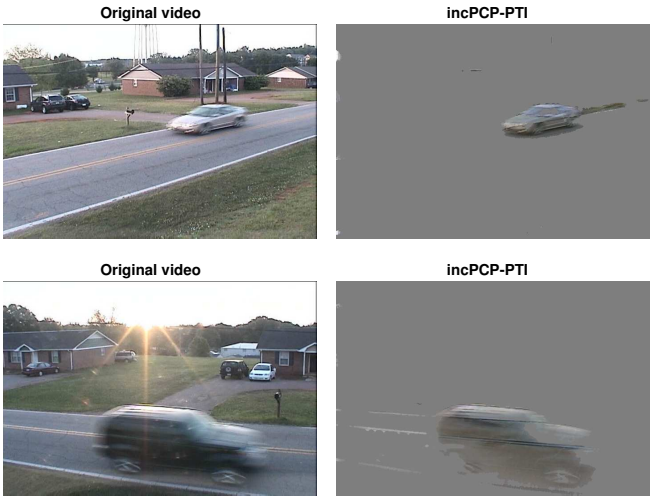


Figure 6. Representative frames of the video and the segmented sparse components for the CP video obtained with both incPCP-PTI and Stab+incPCP-PTI. Frame 90 is shown in the top row and frame 988 is shown in the bottom row.

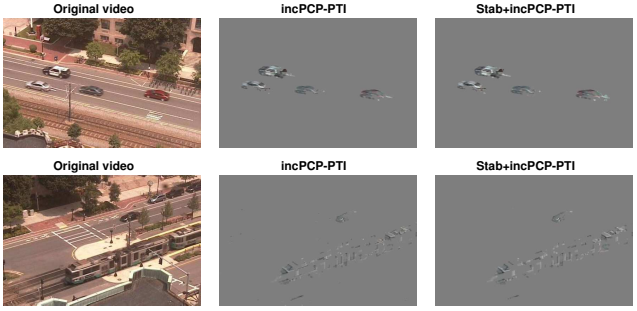


Figure 7. Representative frames of the video and the segmented sparse components for the IP video obtained with both incPCP-PTI and stab+incPCP-PTI. Frame 1260 is shown in the top row and frame 1870 is shown in the bottom row.

time per frame)

## 6. Discussion

It is observed in the results of subsection 5.1.1 that, as expected, the distance $M(\mathbf{s}_k)$ increased, i.e. the sparse approximation was worse, as the panning velocity increased.

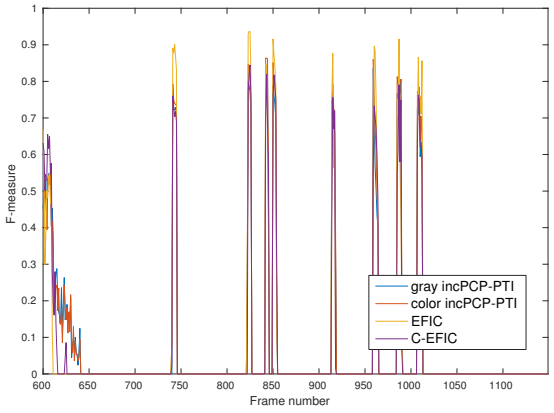


Figure 8. Value of F-measure and computational time per frame for the CP video. NOTE: shown only for available frames (restriction of dataset).

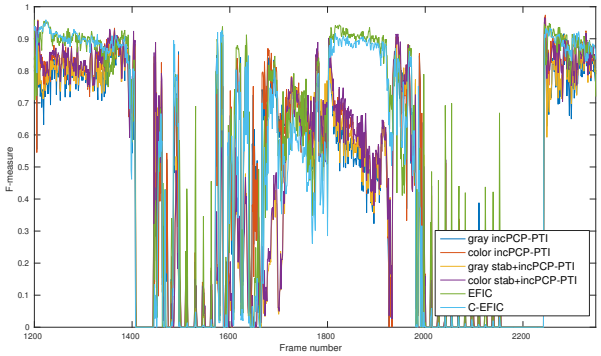


Figure 9. Value of F-measure and computational time per frame for the IP video. NOTE: shown only for available frames (restriction of dataset).

Table 3. Value of F-measure for grayscale and color incPCP-PTI and for EFIC and C-EFIC on the CP video. The F-measure after post-processing with the convex hull is shown in square brackets.

| Method | F-measure | Average time per frame (seconds) |
|---|---|---|
| **grayscale incPCP-PTI** | 0.45 [0.50] | 2.10 |
| **color incPCP-PTI** | 0.47 [0.49] | 3.58 |
| **EFIC** | 0.42 [0.42] | - |
| **C-EFIC** | 0.40 [0.46] | - |

On the contrary, velocity changes did not seem to have a large impact on the sparse estimation. Also expected is the fact that adding jitter to the panning scenario (subsection 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.

Table 4. Value of F-measure for grayscale and color incPCP-PTI and stab+incPCP-PTI on the IP video. The F-measure after post-processing with the convex hull is shown in square brackets.

| Method | Average F-measure | Average time per frame (seconds) |
|---|---|---|
| **grayscale incPCP-PTI** | 0.62 [0.69] | 1.41 |
| **color incPCP-PTI** | 0.64 [0.70] | 2.31 |
| **grayscale stab+incPCP-PTI** | 0.57 [0.63] | (89) + (1.41) |
| **color stab+incPCP-PTI** | 0.59 [0.64] | (89) + (2.31) |
| **EFIC** | 0.68 [0.72] | - |
| **C-EFIC** | 0.64 [0.67] | - |

The inclusion of a video stabilization preprocessing technique (Stab+incPCP-PTI) seemed to decrease such variability 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 2. 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 subsection 5.2 show that incPCP-PTI can perform adequately in real panning videos. The representative frames of Figures 6 and 7 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 9, 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 re-stabilize and perform correctly.

For both the CP and IP videos, 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 suggest 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 post-processing stage. In the IP video, incPCP-PTI + post-processing 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 thorough-out comparison and to draw further conclusion from these comparisons.

## 7. Conclusion

We have presented a novel algorithm, incPCP-PTI, and have shown with artificial datasets and real videos from the CDNET2014 dataset that it can adequately detect moving objects in scenarios with simultaneous panning and jitter. To the best of our knowledge, this is the first PCP like method able to handle the panning conditions. For the synthetic datasets, the algorithm maintained a low distance with respect to the ground truth iALM sparse matrix 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) suggests 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 the incPCP-PTI might be comparable or superior, depending on the case, to state-of-the-art non-PCP like foreground separation methods.

Further improvements of the algorithm might focus on (i) making it able to handle other types of distortions like perspective changes or zooming in/out of the camera, and (ii) reduce the time it takes per frame in order to make it more readily accessible for high frame rate real-time applications.

## 8. Acknowledgment

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

## References

[1] G. Allebosch, F. Deboeverie, P. Veelaert, and W. Philips. EFIC: edge based foreground background segmentation and interior classification for dynamic camera viewpoints. In *Int'l Conf. on Advanced Concepts for Intelligent Vision Systems*. Springer, 2015. 5

[2] 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 *Int'l Joint Conf. on Computer Vision, Imaging and Computer Graphics*, pages 433–454. Springer, 2015. 5

[3] T. Bouwmans, F. Porikli, B. Höferlin, and A. Vacavant. *Background Modeling and Foreground Detection for Video Surveillance*. Chapman and Hall/CRC, 2014. 1

[4] T. Bouwmans, A. Sobral, S. Javed, S. Jung, and E. 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*, 23:1–71, 2017. 1

[5] T. Bouwmans and E. Zahzah. Robust PCA via principal component pursuit: A review for a comparative evaluation in video surveillance. *Computer Vision and Image Understanding*, 122:22–34, 2014. 1, 4

[6] C-EFIC results. goo.gl/ctqmNs. 5

[7] S. Calderara, R. Cucchiara, and A. Prati. A distributed outdoor video surveillance system for detection of abnormal people trajectories. In *ACM/IEEE Int'l Conf. on Distributed Smart Cameras*, pages 364–371, Sept 2007. 1

[8] E. Candès, X. Li, Y. Ma, and J. Wright. Robust principal component analysis? *J. ACM*, 58(3), May 2011. 3

[9] J. Chassery and C. Garbay. An iterative segmentation method based on a contextual color and shape criterion. *IEEE Trans. on PAMI*, (6):794–800, 1984. 5

[10] 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*, 52:410–432, 2016. 2

[11] L. Condat. Fast projection onto the simplex and the $\ell_1$ ball. *Mathematical Programming*, 158(1):575–585, 2016. 3

[12] J. Dong and H. Liu. Video stabilization for strict real-time applications. *IEEE Trans. on Circuits and Sys. for Video Tech.*, 27(4):716–724, April 2017. 5

[13] J. Duchi, S. Shalev-Shwartz, Y. Singer, and T. Chandra. Efficient projections onto the l1-ball for learning in high dimensions. In *ACM ICML*, pages 272–279, 2008. 3

[14] S. Ebadi, V. Ones, and E. Izquierdo. Efficient background subtraction with low-rank and sparse matrix decomposition. In *IEEE ICIP*, pages 4863–4867. IEEE, 2015. 2

[15] EFIC results. goo.gl/LQBeKR. 5

[16] A. Elgammal, D. Harwood, and L. Davis. *Non-parametric Model for Background Subtraction*, pages 751–767. Springer Berlin Heidelberg, Berlin, Heidelberg, 2000. 1

[17] J. Feng, H. Xu, and S. Yan. Online robust pca via stochastic optimization. In *Advances in NIPS*, pages 404–412, 2013. 1

[18] D. Goldfarb and S. Ma. Convergence of fixed-point continuation algorithms for matrix rank minimization. *Foundations of Computational Mathematics*, 11(2):183–210, 2011. 3

[19] J. He, L. Balzano, and A. Szlam. Incremental gradient on the grassmannian for online foreground and background separation in subsampled video. In *IEEE CVPR*, pages 1568–1575, 2012. 1, 2

[20] J. He, D. Zhang, L. Balzano, and T. Tao. Iterative grassmannian optimization for robust image alignment. *Image and Vision Computing*, 32(10):800–813, 2014. 2

[21] Z. Lin, M. Chen, and Y. Ma. The augmented lagrange multiplier method for exact recovery of corrupted low-rank matrices. *arXiv preprint arXiv:1009.5055*, 2010. 1, 4

[22] G. Liu, Z. Lin, and Y. Yu. Robust subspace segmentation by low-rank representation. In *ICML*, pages 663–670, 2010. 1

[23] L. Maddalena and A. Petrosino. A self-organizing approach to background subtraction for visual surveillance applications. *IEEE Trans. on Image Processing*, 17(7):1168–1177, July 2008. 1

[24] 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 Trans. on PAMI*, 34(11):2233–2246, 2012. 2

[25] M. Rahmani and G. Atia. High dimensional low rank plus sparse matrix decomposition. *IEEE Trans. on Signal Processing*, 65(8):2004–2019, 2017. 1

[26] Results for CDnet 2014. goo.gl/SSBvFA. 5

[27] P. Rodríguez. An accelerated newton's method for projections onto the l1-ball. In *IEEE MSLP, 2017*, (Accepted). 3

[28] P. Rodríguez and B. Wohlberg. Fast principal component pursuit via alternating minimization. In *IEEE ICIP*, pages 69–73, Sept. 2013. 1

[29] P. Rodríguez and B. Wohlberg. A matlab implementation of a fast incremental principal component pursuit algorithm for video background modeling. In *IEEE ICIP*, pages 3414–3416, 2014. 1, 2, 3

[30] P. Rodríguez and B. Wohlberg. Translational and rotational jitter invariant incremental principal component pursuit for video background modeling. In *IEEE ICIP*, pages 537–541, 2015. 2, 3

[31] P. Rodríguez and B. Wohlberg. Ghosting suppression for incremental principal component pursuit algorithms. In *IEEE GlobalSIP*, pages 197–201, 2016. 4

[32] P. Rodríguez and B. Wohlberg. Incremental principal component pursuit for video background modeling. *J. of Mathematical Imaging and Vision*, 55(1):1–18, 2016. 1, 2, 3

[33] P. Rodríguez and B. Wohlberg. An incremental principal component pursuit algorithm via projections onto the l1 ball. In *XXIV International Congress of Electrical Engineering, Electronics and Computing*, (Accepted). 3, 4

[34] M. Shah, J. Deng, and B. Woodford. Video background modeling: recent approaches, issues and our proposed techniques. *Machine vision and applications*, 25(5):1105–1119, 2014. 1

[35] N. Srebro, J. Rennie, and T. Jaakola. Maximum-margin matrix factorization. In *Advances in NIPS*, pages 1329–1336. MIT Press, 2005. 1

[36] USC Neovision2 Project. goo.gl/5Si2Nm. 4

[37] Y. Wang, P. Jodoin, F. Porikli, J. Konrad, Y. Benezeth, and P. Ishwar. CDnet 2014: an expanded change detection benchmark dataset. In *IEEE CVPR*, pages 387–394, 2014. 2, 4

[38] J. Wright, A. Ganesh, S. Rao, Y. Peng, and Y. Ma. Robust principal component analysis: Exact recovery of corrupted low-rank matrices via convex optimization. In *Advances in NIPS*, pages 2080–2088, 2009. 1

[39] Y. Xu, J. Dong, B. Zhang, and D. Xu. Background modeling methods in video analysis: A review and comparative evaluation. *CAAI Trans. on Intelligence Technology*, 1(1):43 – 60, 2016. 1

[40] X. Zhou, C. Yang, and W. Yu. Moving object detection by detecting contiguous outliers in the low-rank representation. *IEEE Trans. on PAMI*, 35(3):597–610, 2013. 2

[41] Z. Zivkovic. Improved adaptive gaussian mixture model for background subtraction. In *IEEE ICPR 2004*, volume 2, pages 28–31, 2004. 1