Robust PCA via Principal Component Pursuit: A Review for a Comparative Evaluation in Video Surveillance

Thierry Bouwmans[1], El Hadi Zahzah[2]

[1] *Lab. MIA, Univ. La Rochelle, France*
[2] *Lab. L3i, Univ. La Rochelle, France*
contact email: tbouwman@univ-lr.fr

Abstract

Foreground detection is the first step in video surveillance system to detect moving objects. Recent research on subspace estimation by sparse representation and rank minimization represents a nice framework to separate moving objects from the background. Robust Principal Component Analysis (RPCA) solved via Principal Component Pursuit decomposes a data matrix A in two components such that $A = L + S$, where L is a low-rank matrix and S is a sparse noise matrix. The background sequence is then modeled by a low-rank subspace that can gradually change over time, while the moving foreground objects constitute the correlated sparse outliers. To date, many efforts have been made to develop Principal Component Pursuit (PCP) methods with reduced computational cost that perform visually well in foreground detection. However, no current algorithm seems to emerge and to be able to simultaneously address all the key challenges that accompany real-world videos. This is due, in part, to the absence of a rigorous quantitative evaluation with synthetic and realistic large-scale dataset with accurate ground truth providing a balanced coverage of the range of challenges present in the real world. In this context, this work aims to initiate a rigorous and comprehensive review of RPCA-PCP based methods for testing and ranking existing algorithms for foreground detection. For this, we first review the recent developments in the field of RPCA solved via Principal Component Pursuit. Furthermore, we investigate how these methods are solved and if incremental algorithms and real-time implementations can be achieved for foreground detection. Finally, experimental results on the Background Models Challenge (BMC) dataset which contains different synthetic and real datasets show the comparative performance of these recent methods.

Keywords: Foreground Detection, Robust Principal Component Analysis, Principal Component Pursuit.

1. Introduction

The detection of moving objects is the basic low-level operations in video analysis. This detection is usually done using foreground detection. This basic operation consists of separating the moving objects called ”foreground” from the static information called ”background”. Many foreground detection methods have been developed [8][48][52] and the main resources can be found at the Background Subtraction Web Site[1]. In 1999, Oliver et al. [39] are the first authors to model the background by Principal Component Analysis (PCA). Foreground detection is then achieved by thresholding the difference between the generated background image and the current image. PCA provides a robust model of the probability distribution function of the background, but not of the moving objects while they do not have a significant contribution to the model. Although there are several PCA improvements [56][57] that addressed the limitation of classical PCA with respect to outlier and noise, yielding the field of robust PCA, these methods do not possess the strong performance guarantees provided by very recent works based on the idea that the data matrix A can be decomposed into two components such that $A = L + S$, where L is a low-rank matrix and S is a matrix that can be sparse or not. This decomposition can be obtained by Robust Principal Component Analysis (RPCA) solved via Principal Component Pursuit (PCP) [10][64]. The background sequence is then modeled by a low-rank subspace that can gradually change over time, while the moving foreground objects constitute the correlated sparse outliers. For example, Fig. 1 shows the original frame 309 of the se-

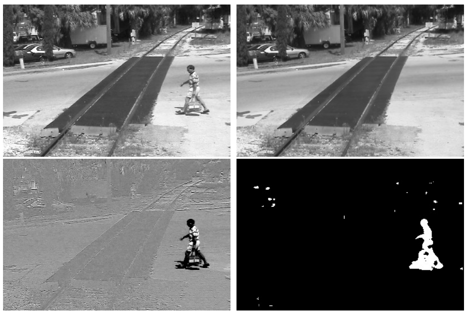

Figure 1: First Row: Original image (309), low-rank matrix L (background), Second Row: Sparse matrix S (foreground), foreground mask.

quence from [49] and its decomposition into the low-rank matrix L and sparse matrix S. We can see that L corresponds to the background whereas S corresponds to the foreground. The fourth image shows the foreground mask obtained by thresholding the matrix S. So, these recent advances in RPCA are fundamental and can be applied to background modeling and foreground detection for video surveillance.

1.1. Related works and issues

Recent advances on subspace estimation by sparse representation and rank minimization show a nice framework to separate moving objects from the background. These advances concern robust subspace tracking and robust PCA models. A representative robust subspace tracking is the online algorithm GRASTA (Grassmannian Robust Adaptive Subspace Tracking Algorithm) [25][26] for low rank subspace tracking, which is robust to both highly incomplete information and sparse corruption by outliers. GRASTA incorporates the augmented Lagrangian of l_1-norm loss function into the Grassmannian optimization framework to alleviate the corruption by outliers in the subspace update at each gradient step. So, GRASTA can estimate and track non-stationary subspaces when the streaming data vectors are corrupted with outliers. GRASTA [26] was evaluated quantitatively with success on background modeling and foreground detection. While GRASTA used the l_1-norm as a convex relaxation of the ideal sparsifying function, Seidel et al. [23][14] recently approach the problem with a smoothed l_p-norm in their algorithm called pROST (l_p-norm Robust Subspace Tracking). Experimental results [14] show that pROST outperforms GRASTA in the case of dynamic backgrounds. For Robust PCA via low-rank and sparse matrix decomposition, several approaches have been developed and can be classified as follows:

- RPCA via Principal Component Pursuit (PCP) [10]
- RPCA via Outlier Pursuit [66]
- RPCA via Iteratively Reweighted Least Squares [21][20][19]
- Bayesian RPCA [13], variational BRPCA [6]
- Approximated RPCA (GoDec [71])

In this paper, we focus only on the review for RPCA-PCP for a systematic evaluation and comparative analysis with the BMC dataset [59].

The first work on RPCA-PCP developed by Candes et al. [10][64] proposed a convex optimization to address the robust PCA problem. Under minimal assumptions, this approach called Principal Component Pursuit (PCP) perfectly recovers the low-rank and the sparse matrices. The background sequence is then modeled by a low-rank subspace that can gradually change over time, while the moving foreground objects constitute the correlated sparse outliers. Candes et al. [10] showed visual results on foreground detection that demonstrated encouraging performance but PCP present several limitations for foreground detection.

The first limitation is that the algorithms to be solved are computational expensive. The second limitation is that PCP is a batch method that stacked a number of training frames in the input observation matrix. In real-time application such as foreground detection, it would be more useful to quickly estimate the low-rank matrix and the sparse matrix in an incremental way for each new frame, rather than as a batch. The third limitation is that the spatial and temporal features are lost, as each frame is represented as a column vector. The fourth limitation is that PCP imposed the low-rank component being exactly low-rank and the sparse component being exactly sparse but the observations such as in video surveillance are often corrupted by noise affecting every entry of the data matrix. The fifth limitation is that PCP assumed that all entries of the matrix to be recovered are exactly known via the observation and that the distribution of corruption should be sparse and random enough without noise. These assumptions are rarely verified in the case in real applications because (1) only a fraction of entries of the matrix can be observed in some environments, (2) the observation can be corrupted by both impulsive and Gaussian noise (3) the outliers i.e moving objects are spatially localized.

[1] https://sites.google.com/site/backgroundsubtraction/Home

Many efforts have been recently concentrated to develop low-computational algorithms for solving PCP [31][32]
[9][70][68][50][64][38][35][33][36][15], to develop incremental algorithms of PCP to update the low-rank and sparse matrix when a new data arrives [41][43][42][44] and real-time implementations [2][3][54]. Moreover, other efforts have addressed problems that appear specifically in real application such as: (1)Presence of noise, (2) Quantization of the pixels, (3) Spatial constraints of the foreground pixels and (4) Local variations in the background. To address (1), Zhou et al. [72] proposed a stable PCP (SPCP) that guarantee stable and accurate recovery in the presence of entry-wise noise. Becker et al. [7] proposed a inequality constrained version of PCP to take into account the quantization error of the pixel's value (2). To address (3), Tang and Nehorai [54] proposed a block-based PCP (BPCP) method via a decomposition that enforces the low-rankness of one part and the block sparsity of the other part. Wohlberg et al. [63] used a decomposition corresponding to a more general underlying model consisting of a union of low-dimensional subspaces for local variation in the background (4). Table 1 shows an overview of the different versions of RPCA-PCP in term of minimization, constraints and convexity. For each method, we investigated in their respective section its robustness, its spatial and temporal constraints, the existence or not of an incremental version, and of a real-time implementation, and of a quantitative evaluation. Furthermore, RPCA-PCP can been extended to the measurement domain, rather than the pixel domain, for use in conjunction with compressive sensing [61][62][27]. Although experiments show that moving objects can be reliably extracted by using a small amount of measurements, we have limited the investigation and the comparative evaluation in this paper to the pixel domain to permit the comparison with the classical background subtraction methods with the BMC dataset [59].

1.2. Motivation and contributions

These recent advances in RPCA via PCP are fundamental and can be applied to foreground detection. However, no algorithm today seems to emerge and to be able to simultaneously address all the key challenges that accompany real-world videos. This is due, in part, to the absence of a rigorous quantitative evaluation with synthetic and realistic large-scale dataset with accurate ground truth providing a balanced coverage of the range of challenges present in the real world. Indeed, the authors usually compared qualitatively their method to RSL[56] or PCP [10]. Some recent quantitative evaluations in foreground detection using the performance metrics have been made but they are limited to one algorithm. For example, Wang et al. [60] studied only RPCA-PCP solved by APG [32]. Xue et al. [67] and Yang [69] evaluated the RPCA-PCP solved by IALM [31] and Guyon et al. [18] adapted the RPCA-PCP with LADMAP [33]. Guyon et al. [17] validated BPCP for foreground detection too. Results show that BPCP gives better performance than PCP. In a recent comparative analysis and evaluation, Guyon et al. [22] compared five algorithms RSL [56], RPCA-PCP solved via EALM [31], RPCA-PCP solved via IALM [31], QPCP [7] and BRPCA [13] with the Wallflower dataset [58], the I2R dataset [30] and Shah dataset [49]. Experimental results show that BRPCA that address spatial and temporal constraints outperforms the other methods. However, this evaluation is limited to five methods and it is not made on large datasets that present a coverage of the range of challenges present in the real world. When considering all of this, we propose to initiate a comprehensive review of RPCA methods solved by PCP for testing and ranking existing algorithms for foreground detection. Contributions of this paper can be summarized as follows:

- **A review regarding RPCA methods solved via PCP:** The original PCP, the Stable PCP, the quantization based version of PCP, the block based version of PCP and the local PCP are reviewed in Section 3 to Section 7. For each method, we investigate how they are solved, and if incremental and real-time versions are available for foreground detection. Furthermore, their advantages and drawbacks are discussed in the case of outliers due to dynamic backgrounds or illumination changes.

- **A systematic evaluation and comparative analysis:** We compare and evaluate nine RPCA methods solved via PCP on a recent dataset which contains various synthetic and realistic videos in Section 8. This datasets is the Background Models Challenge (BMC) dataset^2 [59]. For the evaluation, we have used the quantitative evaluation framework provided by BMC which allows comparison with the state-of-the-art methods.

The rest of this paper is organized as follows. First, we remind in Section 2 the key challenges which concern background and foreground separation in video-surveillance. Then, we review each original method in its section (Section 3 to Section 7). In each case, we survey how each method are solved, and if incremental and real-time versions have been developed for background and foreground separation. Then, the performance evaluation using quantitative metrics over the five datasets is given in Section 8. Finally, we conclude with promising research directions in Section 9.

2. Challenges in Video Surveillance

Three main conditions assures a good functioning of the background subtraction in video surveillance: the camera is fixed, the illumination is constant and the background is static. In practice, several challenges appear and perturb this process. They are the following ones:

- **Noise image:** It is due to a poor quality image source such as images acquired by a web cam or images after compression.
- **Camera jitter:** In some conditions, the wind may cause the camera to sway back and so it cause nominal motion in the sequence. Foreground mask show false detections due to the motion without a robust maintenance mechanism.
- **Camera automatic adjustments:** Many modern cameras have auto focus, automatic gain control, automatic white balance and auto brightness control. These adjustments modify the dynamic in the color levels between different frames in the sequence.
- **Illumination changes:** They can be gradual such as ones in a day in an outdoor scene or sudden such as a light switch in an indoor scene.
- **Bootstrapping:** During the training period, the background is not available in some environments. Then, it is impossible to compute a representative background image.
- **Camouflage:** A foreground objects pixel characteristics may be subsumed by the modeled background. Then, the foreground and the background can be distinguished.
- **Foreground aperture:** When a moved object has uniform colored regions, changes inside these regions may not be detected. Thus, the entire object may not appear as foreground. Foreground masks contain false negative detections.
- **Moved background objects:** Background objects can be moved. These objects should not be considered part of the foreground. Generally, both the initial and the new position of the object are detected without a robust maintenance mechanism.
- **Inserted background objects:** A new background object can be inserted. These objects should not be considered part of the foreground. Generally, the inserted background object is detected without a robust maintenance mechanism.
- **Dynamic backgrounds:** Backgrounds can vacillate and this requires models which can represent disjoint sets of pixel values.
- **Beginning moving object:** When an object initially in the background moves, both it and the newly revealed parts of the background called ”ghost” are detected.
- **Sleeping foreground object:** Foreground object that becomes motionless cannot be distinguished from a background object and then it will be incorporated in the background. How to manage this situation depends on the context. Indeed, in some applications, motionless foreground objects must be incorporated and in others it is not the case.
- **Shadows:** Shadows can be detected as foreground and can come from background objects or moving objects [1].

These challenges have different spatial and temporal properties. In the following sections, we will see how some of these challenges can be tackled by the RPCA models.

3. RPCA via Principal Component Pursuit

Candes et al. [10][64] have proposed a convex optimization to address the robust PCA problem. The observation matrix A is assumed represented as:

$$A = L + S \tag{1}$$

[2] http://bmc.univ-bpclermont.fr/

[3] http://perception.csl.uiuc.edu/matrix-rank/samplecode.html

[4] Available on request by email to the corresponding author

[5] https://sites.google.com/a/istec.net/prodrig/Home/en/pubs

[6] http://lmafit.blogs.rice.edu/

[7] http://www2.ie.psu.edu/aybat/codes.html

[8] http://www.ugcs.caltech.edu/ srbecker/rpca.shtml

Methods	Decomposition	Minimization	Constraints	Convexity
PCP Candes et al. [10]	$A = L + S$	$\min_{L,S} \|\|L\|\|_* + \lambda \|\|S\|\|_1$	$A - L - S = 0$	Yes
SPCP Zhou et al. [72]	$A = L + S + E$	$\min_{L,S} \|\|L\|\|_* + \lambda \|\|S\|\|_1$	$\|\|A - L - S\|\|_F < \delta$	Yes
QPCP Becker et al. [7]	$A = L + S$	$\min_{L,S} \|\|L\|\|_* + \lambda \|\|S\|\|_1$	$\|\|A - L - S\|\|_\infty < 0.5$	Yes
BPCP Tang and Nehorai [54]	$A = L + S$	$\min_{L,S} \|\|L\|\|_* + \kappa(1-\lambda)\|\|L\|\|_{2,1} + \kappa\lambda\|\|S\|\|_{2,1}$	$A - L - S = 0$	Yes
LPCP Wohlberg et al. [63]	$A = AU + S$	$\min_{U,S} \alpha\|\|U\|\|_* + \beta\|\|U\|\|_{2,1} + \beta\|\|S\|\|_1$	$A - AU + S = 0$	Yes

Table 1: RPCA solved via PCP: An Overview

Categories	Solvers	Complexity
Basic algorithms	Singular Value Threshold (SVT[1]) Cai et al. [9]	$O(mnmin(mn))$
	Iterative Thresholding (IT) Wright et al. [64]	$O(mn\min(mn))$
	Accelerated Proximal Gradient (APG)[3]) Lin et al.[32]	$O(mn\min(mn))$ Full SVD
	Dual Method (DM)[3]) Lin et al.[32]	$O(rmn)$ Partial SVD
	Augmented Lagrangian Method (ALM) (EALM[3]) Lin et al.[31]	$O(mn\min(mn))$
	Augmented Direction Method (ADM) (IALM[3]) Lin et al. [31]	$O(rmn)$
	Alternating Direction Method (ADM) (LRSD[2]) Yuan and Yang [70]	$O(mn\min(mn))$
	Symmetric Alternating Direction Method (SADM[4]) Ma [36], Goldfarb et al. [15]	$O(1/\epsilon)$ Iteration complexity
	Non Convex Splitting ADM (NCSADM) Chartrand [12]	Unknown
Linearized algorithms	Linearized Augmented Lagrangian Method (LALM) Yang and Yuan [68]	$O(mn\min(mn))$
	Linearized Alternating Direction Method (LADM) Yang and Yuan [68]	$O(mn\min(mn))$
	Linearized Alternating Direction Method with Adaptive Penalty (LADMAP[4]) Lin et al. [33]	$O(rmn)$ Accelerated version
	Linearized Symmetric Alternating Direction Method (LSADM[4]) Ma [36], Goldfarb et al. [15]	$O(1/\epsilon)$ Iteration complexity
	Fast Linearized Symmetric Alternating Direction Method (Fast-LSADM[4]) Ma [36], Goldfarb et al. [15]	$O(1/\sqrt{\epsilon})$ Iteration complexity
	Linearized Alternating Direction Method (LADM) (LMaFit[6]) Shen et al. [50]	Unknown
Fast algorithms	Randomized Projection for ALM (RPALM) Mu et al. [38]	$O(pmn)$
	l_1 filtering (LF[4]) Liu et al. [35]	$O(r(m+n))$
	Block Lanczos with Warm Start Lin and Wei [34]	Unknown
	Fast Alterning Minimization (FAM)[5] Rodriguez and Wohlberg [47]	Unknown

Table 2: Algorithms for solving RPCA-PCP: An Overview

Methods	Solvers	Complexity
RPCA-SPCP	Alternating Splitting Augmented Lagrangian method (ASALM)	Unknown
Zhou et al. [72]	Tao and Huan [55]	
	Variational ASALM (VASALM)[3]	Unknown
	Tao and Huan [55]	
	Parallel ASALM (PSALM)	Unknown
	Tao and Huan [55]	
	Non Smooth Augmented Lagrangian Algorithm (NSA[7])	$O(1/\epsilon)$
	Aybat et al. [4]	Iteration complexity
	Partially Smooth Proximal Gradient (PSPG[7])	$O(1/\epsilon)$
	Aybat et al. [5]	Iteration complexity
RPCA-QPCP)	Templates for First-Order Conic Solvers (TFOCS[8])	$O(1/\epsilon)$
Becker et al. [7]	Becker et al. [7]	Iteration complexity
RPCA-BPCP	Augmented Lagrangian Method (ALM)	$O(mn \min(mn))$
(RPCA-LBD[3]) Tang and Nehorai [54]	Tang and Nehorai [54]	
RPCA-LPCP	Split Bregman Algorithm (SBA)	Unknown
Wohlberg et al. [63]	Goldstein and Osher [16]	

Table 3: Algorithms for solving RPCA: An Overview

where L is a low-rank matrix and S must be sparse matrix with a small fraction of nonzero entries. The straightforward formulation is to use l_0-norm to minimize the energy function:

$$\min_{L,S} \ rank(L) + \lambda ||S||_0 \quad \text{subj} \quad A - L - S = 0 \qquad (2)$$

where $\lambda > 0$ is an arbitrary balanced parameter. But since this problem is NP-hard, a typical solution would involve a search with combinatorial complexity. This research seeks to solve for L with the following optimization problem:

$$\min_{L,S} \ ||L||_* + \lambda ||S||_1 \quad \text{subj} \quad A - L - S = 0 \qquad (3)$$

where $||.||_*$ and $||.||_1$ are the nuclear norm (which is the l_1-norm of singular value) and l_1-norm, respectively, and $\lambda > 0$ is an arbitrary balanced parameter. Usually, $\lambda = \frac{1}{\sqrt{\max(m,n)}}$. Under these minimal assumptions, this approach called Principal Component Pursuit (PCP) solution perfectly recovers the low-rank and the sparse matrices. Candes et al. [72] showed results on face images and background modeling that demonstrated encouraging performance.

Background modeling and RPCA-PCP: The low-rank minimization concerning L offers a suitable framework for background modeling due to the correlation between frames. So, minimizing L and S implies that the background is approximated by a low-rank subspace that can gradually change over time, while the moving foreground objects constitute the correlated sparse outliers which are contained in S. To obtain the foreground mask, S needs to be thresholded (See Fig. 1). The threshold is determined experimentally. $rank(L)$ influences the number of modes of the background that can be represented by L: If $rank(L)$ is to high, the model will incorporate the moving objects in its representation; if the $rank(L)$ is to low, the model tends to be uni-modal and then the multi-modality which appears in dynamic backgrounds will be not captured. The quality of the background/foreground separation is directly related to the assumption of the low-rank and sparsity of the background and foreground, respectively. In this case, the best separation is then obtained only when the optimization algorithm has converged. Practically, RPCA-PCP present several limitations developed in the Section 1.1 and an overview of their solutions is given in the following sections.

3.1. Algorithms for solving PCP

Several algorithms complexity have been proposed for solving PCP. An overview of these algorithms as well as their complexity can be seen in Table 2. All these algorithms require solving the following type of subproblem in each iteration:

$$\min_{L,S} \ \eta ||L||_* + \lambda ||S||_F^2 \qquad (4)$$

The above problem can have a closed form solution or not following the application.

- In the first case, PCP can be reformulated as a semidefinite program and then be solved by standard interior point methods [11]. However, interior point methods have difficulty in handling large matrices because the complexity of computing the step direction is $O((mn\min(m,n))^2)$, where $m \times n$ is the size of the data matrix. If $m = n$, then the complexity is $O(n^6)$. So the generic interior point solvers are too limited for many real applications where the number of data are very large. To overcome the scalability issue, only the first-order information can be used. Cai et al. [9] showed that this technique, called Singular Value Thresholding (SVT), can be used to minimize the nuclear norm for matrix completion. As the matrix recovery problem in Equation (3) needs minimizing a combination of both the l_1-norm and the nuclear norm, Wright et al. [64] adopted a iterative thresholding technique (IT) to solve it and obtained similar convergence and scalability properties than interior point methods. However, the iterative thresholding scheme converges extremely slowly. To alleviate this slow convergence, Lin et al. [32] proposed two algorithms: the accelerated proximal gradient (APG) algorithm and the gradient-ascent algorithm applied to the dual of the problem in Equation (3). However, these algorithms are all the same to slow for real application. More recently, Lin et al. [31] proposed two algorithms based on augmented Lagrange multipliers (ALM). The first algorithm is called Exact ALM (EALM) method that has a Q-linear convergence speed, while the APG is in theory only sub-linear. The second algorithm is an improvement of the EALM that is called Inexact ALM (IALM) method, which converges practically as fast as the EALM, but the required number of partial SVDs is significantly less. The IALM is at least five times faster than APG, and its precision is also higher [31]. However, the direct application of ALM treats the Equation (3) as a generic minimization problem and ignores its separable structure emerging in both the objective function and the constraint [31]. Hence, the variables S and L are minimized simultaneously. Yuan and Yang [70] proposed to alleviate this ignorance by the Alternating Direction Method (ADM) which minimizes the variables L and S serially. ADM achieves it with less computation cost than ALM. Recently, Chartrand [12] proposed a non convex splitting version of the ADM [31] called NCSADM . This non convex generalization of [31] produces a sparser model that is better able to separate moving objects and stationary objects. Furthermore, this splitting algorithm maintains the background model while removing substantial noise, more so than the convex regularization does.

- In the second case when the resulting subproblems do not have closed-form solutions, Yang and Yuan [68] proposed to linearize these subproblems such that closed-form solutions of these linearized subproblems can be easily derived. Global convergence of these Linearized ALM (LALM) and ADM (LADM) algorithms are established under standard assumptions. Recently, Lin et al. [35] improved the convergence for the Linearized Alternating Direction Method with an Adaptive Penalty (LADMAP). They proved the global convergence of LADM and applied it to solve Low-Rank Representation (LRR). Furthermore, the fast version of LADMAP reduces the complexity $O(mnmin(m,n))$ of the original LADM based method to $O(rmn)$, where r is the rank of the matrix to recover, which is supposed to be smaller than m and n. In a similar way, Ma[36] and Goldfarb et al. [15] proposed a Linearized Symmetric Alternating Direction Method (LSADM) for minimizing the sum of two convex functions. This method requires at most $O(1/\epsilon)$ iterations to obtain an ϵ-optimal solution, and its fast version called Fast-LSADM requires at most $O(1/\sqrt{\epsilon})$ with little change in the computational effort required at each iteration.

All these methods require computing SVDs for some matrices, resulting in $O(mn\min(m,n))$ complexity. Although partial SVDs are used to reduce the complexity to $O(rmn)$ such a complexity is still high for large data sets. Therefore, recent researches focus on the reduction of the complexity by avoiding computation of SVD. Shen et al. [50] presented a method where the low-rank matrix is decomposed in a product of two-low rank matrices and then minimized over the two matrices alternatively. Although, they do not require nuclear norm minimization and so the computation of SVD, the convergence of the algorithm is not guaranteed as the problem is non-convex. Furthermore, both the matrix-matrix multiplication and QR decomposition based rank estimation technique require $O(rmn)$ complexity. So, this method does not essentially reduce the complexity. In another way, Mu et al. [38] reduced the problem scale by random projections but different random projections may lead to radically different results. Furthermore, additional constraint to the problem slows down the convergence. The complexity of this method is also $O(pmn)$ where $p \times m$ is the size of the random projection matrix. So, this method is still nor linear complexity with respect to the matrix size. Recently, Liu et al. [46] proposed a novel algorithm, called l_1-filtering for exactly solving PCP with an $O(r^2(m+n))$ complexity. This method is a truly linear cost method to solve PCP problem when the date size is very large while the target rank is small. Moreover, l_1-filtering is highly parallelizable. It is the first algorithm that can exactly solve a nuclear norm minimization problem in linear time. Numerical experiments [46] show the great performance of l_1-filtering in speed compared to the previous algorithms for solving PCP.

3.2. Algorithms for incremental PCP

PCP is an offline method which treats each image frame as a column vector of the matrix A. In real-time application such as foreground detection, it would be more useful to estimate the sparse matrix in an incremental way quickly as new frame comes rather than in batch way. Furthermore, the sparsity structure may change slowly or in a correlated way, which may result in a low-rank sparse matrix. In this case, PCP assumption will not get satisfied and S can't be separated from L. Moreover, the principal directions can change over time. So, the rank of the matrix L will keep increasing over time thus making PCP infeasible to do after time. This last issue can be solved by not using all frames of the sequences. To address the two first issues, Qiu and Vaswani [41] proposed an online approach called Recursive Robust PCP (RR-PCP[7]) [41] and Recursive Projected Compressive Sensing (ReProCS[7]) [42]. The aim of RR-PCP (ReProCS) is to causally keep updating the sparse matrix S_t at each time, and keep updating the principal directions. The t-th column of A, A_t, is the data acquired at time t and can be decomposed as follows:

$$A_t = L_t + S_t = [UI]\,[x_t S_t]^t \tag{5}$$

where $x_t = U^T L_t$ and the matrix U is an unknown $m \times m$ orthonormal matrix. The support of S_t changes slowly over time. Let N_t denote the support of x_t which is assumed piecewise constant over time and given an initial estimate of $P_t = (U)_{N_t} = \hat{P}_t$, Qiu and Vaswani [41] solved for the sparse component S_t by finding the orthogonal complement matrix $\hat{P}_{t,\perp}$, and then using the projection M_t onto $\hat{P}_{t,\perp}$, denoted by y_t:

$$y_t = \hat{P}^T_{t,\perp} M_t = \hat{P}^T_{t,\perp} L_t + \hat{P}^T_{t,\perp} S_t \tag{6}$$

to solve S_t. The low-rank component is close to zero if $\hat{P}_t \approx P_t$, otherwise new directions are added. Furthermore, recent estimates of $L_t = A_t - S_t$ are stored and used to update P_t. RR-PCP requires the support x_t to be fixed and quite small for a given support size S_t, but this does often not hold. So, RR-PCP could not handle large outliers support sizes. Qiu and Vaswani [43] address this problem by using time-correlation of the outliers. This method called Support-Predicted Modified-CS RR-PCP[8] [43] and Support-Predicted Modified-CS[8] [42] is also an incremental algorithm and outperforms the RR-PCP. However, this algorithm is only adapted for specific situation where there are only one or two moving objects that remain in scene. But, this is not applicable to real videos where multiple and time-varying number of objects can enter of leave the scene. Moreover, it requires knowledge of foreground motion. Recently, Qiu and Vaswani [44] proposed a method called automated Recursive Projected CS (A-ReProCS[8]) that ensures robustness when there are many nonzero foreground pixels, that is, there are many moving objects or large moving objects. Furthermore, A-ReProCS outperforms the previous incremental algorithms when foreground pixels are correlated spatially or temporally and when the foreground intensity is quite similar to the background one.

3.3. Methods for real time implementation of PCP

Despite the efforts to reduce the time complexity, the corresponding algorithms have prohibitive computational time for real application such as foreground detection. The main computation in PCP is the singular value decomposition of the large matrix A. Instead of computing a large SVD on the CPU, Anderson et al. [2][3] proposed an implementation of the communication-avoiding QR (CAQR) factorization that runs entirely on the GPU. This implementation achieved 30× speedup over the implementation on CPU using MKL.

Recently, Pope et al.[40] proposed a variety of methods that significantly reduce the computational time of the ALM algorithm. These methods can be classified as follows:

- **Reducing the computation time of SVD:** The computation of the SVD is reduced using the Power method [40] that enables to compute the singular values in a sequential manner, and to stop the procedure when a singular value is smaller than a threshold. The use of the Power method by itself results in 4.32× lower runtime. Furthermore, the gain is improved by a factor of 2.02× speedup if the procedure is stopped when the singular value is

[8] http://home.engineering.iastate.edu/ chenlu/ReProCS/

smaller than the threshold. If the rank of L is fixed and the Power SVD is stopped when the number of singular value is equal to $rank(L)$, the additional speedup is 17.35. On the results, the threshold fixed to $rank(L)$ influences the number of modes of the background that can be represented by L.

- **Seeding the PCP algorithm:** PCP operates on matrices consisting of blocks of contiguous frames acquired with a fixed camera. So, the low-rank matrix does not change significantly from one block to the next. Thus, Pope et al. [40] use the low-rank component obtained by the ALM algorithm from the previous block as a starting point for the next block. This method allows an additional speedup of 7.73.
- **Partioning into subproblems:** Pope et al. [40] proposed to partition the matrix A into P smaller submatrices. The idea is to combine the solutions of the P corresponding PCP subproblems to recover the solution of the full matrix A at lower computational complexity.

These methods showed a speedup of the ALM algorithm by more than 365 times compared to a C implementation.

3.4. Methods for finding the optimal PCP solution

PCP recovers the true underlying low-rank matrix when a large portion of the measured matrix is either missing or arbitrarily corrupted. However, in the absence of a true underlying signal L and the deviation S, it is not clear how to choose a value of λ that produces a good approximation of the given data A for a given application. A typical approach would involve some cross-validation step to select λ to maximize the final results of the application. The issue with cross-validation in this situation is that the best model is selected indirectly in terms of the final results, which can depend in unexpected ways on later stages in the data processing chain of the application. Instead, Ramirez and Sapiro [45][46] addressed this issue via the Minimum Description Length (MDL) principle [24] and so proposed a MDL-based low-rank model selection. The principle is to select the best low-rank approximation by means of a direct measure on the intrinsic ability of the resulting model to capture the desired regularity from the data. To obtain the family of models $\mathbf{M}$ corresponding to all possible low-rank approximation of A, Ramirez and Sapiro [45][46] applied the RPCA decomposition for a decreasing sequence of values of λ, $\{\lambda_t : t = 1, 2, 3, ...\}$ obtaining a corresponding sequence of decomposition $\{(L_t, S_t), t = 1, 2, 3, ...\}$. This sequence is obtained via a simple modification of the ALM algorithm [31] to allow warm restarts, that is, where the initial ALM iterate for computing (L_t, S_t) is (L_{t-1}, S_{t-1}). Finally, Ramirez and Sapiro [45][46] select the pair $(L_{\hat{t}}, S_{\hat{t}})$,$\hat{t} = \arg\min_t \{MDL(L_t) + MDL(S_t)\}$ where $MDL(L_t) + MDL(S_t) = MDL(A|M)$ denoted the description length in bits of A under the description provided by a given model $M \in \mathbf{M}$. Experimental results show that the best λ is not the one determined by the theory in Candes et al. [10].

4. RPCA via Stable Principal Component Pursuit

PCP is limited to the low-rank component being exactly low-rank and the sparse component being exactly sparse but the observations in real applications are often corrupted by noise affecting every entry of the data matrix. Therefore, Zhou et al. [72] proposed a stable PCP (SPCP) that guarantee stable and accurate recovery in the presence of entry-wise noise. So, this measurement model assumes that the observation matrix A is represented as follows:

$$A = L + S + E \tag{7}$$

where E is a noise term (say i.i.d. noise on each entry of the matrix) and $||E||_F < \delta$ for some $\delta > 0$. To recover L and S, Zhou et al. [72] proposed to solve the following optimization problem, as a relaxed version to PCP:

$$\min_{L,S} \ ||L||_* + \lambda ||S||_1 \quad \text{subj} \quad ||A - L - S||_F < \delta \tag{8}$$

where $||.||_F$ is the Frobenius norm and $\lambda = \frac{1}{\sqrt{n}}$.

4.1. Algorithms for solving SPCP

Just as in Equation (2) for PCP, Tao and Huan [55] showed that an easy reformulation of the constrained convex programming for Equation (8) falls perfectly in the applicable scope of the classical ALM. Moreover, the favorable separable structure emerging in both the objective function and the constraints entails the idea of splitting the corresponding augmented Lagrangian function to derive efficient numerical algorithms. So, Tao and Huan [55] developed the alternating splitting augmented Lagrangian method (ASALM) and its variant (VASALM), and the parallel splitting augmented Lagrangian method (PSALM) for solving Equation (8). ASALM and its variants converge to an optimal solution. However, ASALM iterations are too slow for real time application and its complexity is not known.

To address this problem, Aybat et al. [4] studied how fast first-order methods can be applied to SPCP with low complexity iterations and showed that the subproblems that arise when applying optimal gradient methods of Nesterov, alternating linearization methods and alternating direction augmented Lagrangian methods to the SPCP problem either have closed-form solutions or have solutions that can be obtained with very modest effort. Furthermore, Aybat et al. [4] developed a new first order algorithm called Non Smooth augmented Lagrangian Algorithm (NSA), based on partial variable splitting. All but one of the methods analyzed require at least one of the non-smooth terms in the objective function to be smoothed and obtain an ϵ-optimal solution to the SPCP problem in $O(\sqrt{\epsilon})$ iterations. NSA, which works directly with the fully non-smooth objective function, is proved to be convergent under mild conditions on the sequence of parameters it uses. NSA, although its complexity is not known, is the fastest among the optimal gradient methods, alternating linearization methods and alternating direction augmented Lagrangian methods algorithms and substantially outperforms ASALM. Recently, Aybat et al. [5] proposed a proximal gradient algorithm called Partially Smooth Proximal Gradient (PSPG). Experimental results show that both the number of partial SVDs and the CPU time of PSPG are significantly less than those for NSA and ASALM. An overview of these algorithms as well as their complexity can be seen in Table 3.

4.2. *Methods for real time implementation of SPCP*

Mackey et al. [37] proposed a real time implementation framework, entitled Divide-Factor-Combine (DFC). DFC randomly divides the original matrix factorization task into cheaper subproblems in term of complexity, solves those subproblems in parallel using any base matrix factorization (MF) algorithm, and combines the solutions to the subproblem using an efficient technique from randomized matrix approximation. The inherent parallelism of DFC allows for near-linear to superlinear speedups in practice, while the theory provides high-probability recovery guarantees for DFC comparable to those provided by its base algorithm. So, Mackey et al. [37] proposed two algorithms called DFC-PROJ and DFC-NYS, that differ from the method used to divide the original matrix. DFC-PROJ randomly partitions the orthogonal projection of the matrix A onto the t l-column submatrices $C_1, ..., C_t$ by using a column sampling method, while DFC-NYS selects an l-column submatrix C and an d-row submatrix R using the generalized Nystrm method. DFC significantly reduces the per-iteration complexity to $O(mlr_{C_1})$ where r_{C_1} is the rank of the matrix C_1 for the DFC-PROJ. The cost of combining the submatrix estimates is even smaller, since the outputs of standard MF algorithms are returned in factored form. Indeed, the column projection step of DFC-PROJ requires only $O(mr^2 + lr^2)$ time for $r = \max_i k_{C_i}$, $O(mr^2 + lr^2)$ time for the pseudoinversion of C_i and $O(mr^2 + lr^2)$ time for matrix multiplication with each C_i in parallel. For the DFC-NYS, the per-iteration complexity $O(mlr_C)$ where r_C is the rank of the matrix C and $O(mlr_R)$ where r_R is the rank of the matrix R. The cost of combining requires $O(l\bar{r}^2 + d\bar{r}^2 + \min(m, n)\bar{r}^2)$ time where $\bar{r} = \max(r_C, r_R)$. Mackey et al. [37] improved these real time implementations by using ensemble methods that have been shown to improve performance of matrix approximation algorithms, while straightforwardly leveraging the parallelism of modern many-core and distributed architectures [28]. As such, an ensemble variants of the DFC algorithms have been developed reducing recovery error while introducing a negligible cost to the parallel running time. For DFC-PROJ-ENS, rather than projecting only onto the column space of C_1, the projection of $C_1, ..., C_t$ is done onto the column space of each C_i in parallel and then average the t resulting low-rank approximations.For DFC-NYS-ENS, a random d-row submatrix is chosen as in DFC-NYS and independently partition the columns of the matrix in l as in DFC-PROJ. After running the base MF algorithm on each submatrix, the generalized Nystrom method is applied to each pair of matrices in parallel and then the t resulting low-rank approximations is obtained by average.

5. RPCA via quantization-based Principal Component Pursuit

Becker et al. [7] proposed a inequality constrained version of RPCA proposed by Candes et al. [10] to take into account the quantization error of the pixel's value. Indeed, each pixel has a value between $0, 1, 2, \ldots, 255$. This value is the quantized version of the real value which is between $[0, 255]$. So, the idea is to apply RPCA to the real observations instead of applying it to the quantized ones. Indeed, it is unlikely that the quantized observation can be split nicely into a low-rank and sparse component. So, Becker et al. [7] supposed that $L + S$ is not exactly equal to A, but rather that $L + S$ agrees with A up to the precision of the quantization. The quantization can induce at most an error of 0.5 in the pixel value. This measurement model assumes that the observation matrix A is represented as follows:

$$A = L + S + Q \qquad (9)$$

where Q is the error of the quantization. Then, the objective function is the same than the equality version in Equation (3), but instead of the constraints $L + S = A$, the constraints are $||A - L - S||_\infty \leq 0.5$. So, the quantization based PCP (QPCP) is formulated as follows:

$$\min_{L,S} \; ||L||_* + \lambda||S||_1 \quad \text{subj} \quad ||A - L - S||_\infty \leq 0.5 \qquad (10)$$

The ℓ_∞-norm allows to capture the quantization error of the observed value of the pixel.

Algorithms for solving QPCP: Becker et al. [7] used a general framework for solving this convex cone problem called Templates for First-Order Conic Solvers (TFOCS). First, this approach determines a conic formulation of the problem and then its dual. Then, Becker et al. [7] applied smoothing and solved the problem using an optimal first-order method. This approach allows to solve the problem in compressed sensing.

6. RPCA via block-based Principal Component Pursuit

Tang and Nehorai [54] proposed a block-based PCP (BPCP) that enforces the low-rankness of one part and the block sparsity of the other part. This decomposition involves the same model than PCP in Equation (1), that is $A = L + S$, where L is the low-rank component but S is a block-sparse component. The low-rank matrix L and the block-sparsity matrix S can be recovered by the following optimization problem [71]:

$$\min_{L,S} \; ||L||_* + \kappa(1-\lambda)||L||_{2,1} + \kappa\lambda||S||_{2,1}$$

$$\text{subj} \quad A - L - S = 0 \qquad (11)$$

where $||.||_*$ and $||.||_{2,1}$ are the nuclear norm and the $l_{2,1}$-norm, respectively. The $l_{2,1}$-norm corresponds to the l_1-norm of the vector formed by taking the l_2-norms of the columns of the underlying matrix. The term $\kappa(1 - \lambda)||L||_{2,1}$ ensures that the recovered matrix L has exact zero columns corresponding to the outliers. In order to eliminate ambiguity, the columns of the low-rank matrix L corresponding to the outlier columns are assumed to be zeros.

Algorithm for solving BPCP: Tang and Nehorai [60] designed an efficient algorithm to solve the convex problem in Equation (11) based on the ALM method. This algorithm allow to decompose the matrix A in a low-rank and block-sparse matrices in respect to the $||.||_{2,1}$ and the extra term $\kappa(1 - \lambda)||L||_{2,1}$.

7. RPCA via Local Principal Component Pursuit

PCP is highly effective but the underlying model is not appropriate when the data are not modeled well by a single low-dimensional subspace. Wohlberg et al. [63] proposed a decomposition corresponding to a more general underlying model consisting of a union of low-dimensional subspaces.

$$A = AU + S \qquad (12)$$

This idea can be implemented as the following problem:

$$\min_{U,S} \; \alpha||U||_* + \beta||U||_{2,1} + \gamma||S||_1$$

$$\text{subj} \quad A - AU - S = 0 \qquad (13)$$

The explicit notion of low rank, and its nuclear norm proxy, is replaced by representability of a matrix as a sparse representation on itself. The $l_{2,1}$-norm encourages rows of U to be zero, but does not discourage nonzero values among the entries of a nonzero row. The l_1-norm encourages zero values within each nonzero row of U.

To better handle noisy data, Wohlberg et al. [63] modified the Equation (13) with a penalized form and add a Total Variation penalty on the sparse deviations for contiguous regions as follows:

$$\min_{U,S} \; \frac{1}{2}||A - DU - S||_* + \alpha||U||_*$$

$$+\beta||U||_{2,1} + \beta||S||_1 + \delta||grad(S)||_1 \quad \text{subj} \quad A - DU - S = 0 \qquad (14)$$

where the dictionary D is derived from the data A by mean subtraction and scaling, and $grad(S)$ is a vector valued discretization of the 3D gradient of S. An appropriate sparse U can be viewed as generating a locally low-dimensional approximation DU of $A - S$. When the dictionary is simply the data (i.e., $D = A$), the sparse deviations (or outliers) S are also the deviations of the dictionary D, so constructing the locally low-dimensional approximation as $(D - S)U$, implying an adaptive dictionary $D - S$, should allow U to be even sparser.

Algorithm for solving LPCP: Wohlberg et al. [63] proposed to solve the Equation (13) using the Split Bregman Algorithm (SBA) [16]. Adding terms relaxing the equality constraints of each quantity and its auxiliary variable, Wohlberg et al. [63] introduced Bregman variables in the Equation (13). So, the problem is split into an alternating minimization of five subproblems. Two subproblems are l_2 problems that are solved by techniques for solving linear systems such as conjugate gradient. The other three subproblems are solved very cheaply using shrinkage, i.e generalized shrinkage and soft shrinkage.

8. Experimental Evaluation

We have evaluated and compared nine RPCA algorithms based on PCP with the dataset provided for the Background Models Challenge (BMC). We have chosen the most representative algorithms: 1) PCP solved via seven different algorithms (EALM [31], IALM [31], ADM (LRSD) [70], LADMAP [33], LSADM [15], LADM (LMaFit) [50], BLWS [34]), 2) SPCP solved via NSA [4] and 3) QPCP solved via TFOCS [7].

8.1. BMC dataset: A Brief Description

The BMC (Background Models Challenge) dataset consists of both synthetic and real videos to permit a rigorous comparison of background subtraction techniques for the corresponding workshop organized within Asian Conference in Computer Vision (ACCV). This dataset consists of the following sequences:

- **Synthetic Sequences:** A set of 20 urban video sequences rendered with the SiVIC simulator. With this tool, the associate ground truth was rendered frame by frame for each video at 25 fps. Several complex scenarios are simulated such as fog, sun and acquisition noise for two environments (a rotary and a street). A first part of 10 synthetic videos are devoted to the learning phase, while the 10 others are used for the evaluation.
- **Real Sequences:** The BMC dataset also contains 9 real videos acquired from static cameras in video-surveillance contexts for evaluation. This real dataset has been built in order test the algorithms reliability during time and in difficult situations such as outdoor scenes. So, real long videos about one hour and up to four hours are available, and they may present long time change in luminosity with small density of objects in time compared to previous synthetic ones. Moreover, this real dataset allows to test the influence of some difficulties encountered during the object extraction phase, as the presence of vegetation, cast shadows or sudden light changes in the scene.

All the videos were processed with a unique sets of parameters which are tuned based on the synthetic videos for the learning phase.

8.2. Analysis of the Experimental Results

To evaluate the effectiveness of the nine RPCA approaches based on PCP, we compared their performance with the following approaches: 1) original PCA [39], 2) two Bayesian RPCA algorithms (BRPCA [13], VBRPCA [6]), 3) two approximated RPCA algorithms (GoDec [71], SemiSoft GoDec [71]) and 4) the adaptive MOG [53]. Table 6 and Table 7 show the evaluation results using the synthetic videos and the ones using the real videos for evaluation phase, respectively. These tables are divided in four parts which correspond to a type of algorithm: 1) RPCA based on PCP (RPCA-PCP, RPCA-SPCP, RPCA-QPCP and RPCA-BPCP), 2) Bayesian RPCA, 3) Approximated RPCA and 4) Adaptive MOG. First , we provide a short qualitative analysis in presence of illumination changes and dynamic backgrounds. Then, we give a full quantitative evaluation.

Due the page limitation, we present visual results on three video sequences: sequence Rotary 122 (Fig. 2), sequence ”Wandering students” (Fig. 3) and sequence ”Traffic during windy day” (Fig. 4). More visual results are available in the supplementary materials. For the sequence ”Wandering students” (Fig. 3), some gradual illumination changes occur and we can note that the results are visually acceptable. From Fig. 4, we can see that the RPCA model is robust to waving trees but less to the presence of the clouds. As the visual results seem very similar, we have done a full quantitative evaluation.

From Table 6, we can see that the algorithm LSADM gives the best performance for the sequences 112, 212, 312 and 412. TFOCS seems to emerge by giving the best results for the sequence 512, 122, 222 and 522. Furthermore, TFOCS obtains the best average F-Measure against all the other methods. From Table 7, the algorithm NSA outperforms the other PCP algorithms for the sequence 002, 003, 004 and 006. LSADM confirms here its good robustness by obtaining the second average F-Measure after the BLWS. Table 8 groups the F-measure for the best algorithm in each category. The LSADM gives the best F-measure for the synthetic videos. For the real videos, the best performance is obtained by the TFOCS followed by the LSADM. In conclusion, the LSADM and TFOCS are the most robust algorithms follows by the BLWS and NSA. However, these two algorithms are neither real-time and incremental. Another main conclusion is that most of the RPCA algorithms outperforms the adaptive MOG for the synthetic videos as for real videos too.

8.3. Implementation and Computational Cost

All the algorithms are implemented in matlab. The computational cost of the RPCA-PCP algorithms is mainly related to the singular value decomposition (SVD). It can be reduced significantly by using a partial SVD because only the first largest few singular values are needed. Practically, the implementation available in PROPACK[9] are used for the IALM, LADMAP, LSADM and LADM. The SVDs and CPU time of each algorithm were computed for each sequence. Table 4 and Table 5 summarize the average times. The CPU times are reported in the form hh:mm:ss for images of size 144 × 176 and with 200 frames for the training to allow easy comparison with other RPCA algorithms as the previous publications in this field present these performances on the I2R dataset [29] in this data format. In this paper, the results for the ASALM [55] and the PSPG [5] come from their authors. We can see that EALM and ADM are very computational expensive due to the fact that theses algorithms compute full SVDs. On these problems of extremely low ranks, the partial SVD technique used in IALM, LADMAP, LSADM and LADM become quite effective and reduce significantly the computation time. For the SPCP, the PSPG solver is the most efficient follows by the NSA and the ASALM. The variational BRPCA is less computational expensive than the BRPCA. Finally, the GoDec algorithm is the one which requires less time computation time, and then it makes large videos applications reachable in real-time.

9. Conclusion

In this paper, we have firstly presented a review of recent advances on RPCA. We classified them in the following categories: Principal Component Pursuit (PCP), stable PCP, quantization-based PCP, block-based PCP and local PCP. Furthermore, we investigated how these methods are solved and if incremental algorithms and real-time implementations can be achieved for foreground detection. Finally, experimental results on the Background Models Challenge (BMC) dataset show the comparative performance of these recent methods. In conclusion, this review for a comparative evaluation of RPCA-PCP algorithms highlights the following points:

- Batch algorithms such as EALM and ADM (LRSD) are too computational expensive. As, their computational cost is mainly related to the singular value decomposition (SVD), it can be reduced significantly by using a partial SVD such as in IALM, LADMAP, LSADM and LADM.
- There are two incremental versions for PCP, that are ReProCS[42] and AutoReproCS[44] but these methods are not real-time. In another way, there are real-time implementations for PCP and SPCP, but these implementations are for the batch algorithm. So, there are no methods that is both incremental and real-time.
- The comparative evaluation shows that the RPCA models outperform state-of-the-art algorithms such as the MOG algorithms [53][51]. Furthermore, LSADM[15] and TFOCS[7] solvers seem to be the most adapted ones in the field of video surveillance.

Future research may concern less computational SVD algorithms such as LMSVD [65] for batch RPCA, and RPCA-PCP model which would be both incremental and real-time to reach the performance of the state-of-the-art algorithms such as MOG in term of computation time and memory requirements. Finally, RPCA shows a nice potential for background modeling and foreground detection in video surveillance. Furthermore, RPCA can been extended to the measurement domain, rather than the pixel domain, for use in conjunction with compressive sensing.

10. Acknowledgment

First, the authors would like to acknowledge with their sincere gratitude Brendt Wohlberg (Los Alamos National Laboratory, USA) for his collaboration in the first lecture and his hepful comments to improve this paper. The authors would like to thank the following researchers too: 1) Zhouchen Lin (Visual Computing Group, Microsoft Research Asia) who has kindly provided the solver LADMAP [33], 2) Shiqian Ma (Institute for Mathematics and Its Applications, Univ. of Minnesota, USA) who has kindly provided the solver LSADM [36] and 3) Charles Guyon (Lab. MIA, Univ. La Rochelle, France) who has kindly help us for the experimentation.

References

[1] A. Amato, I. Huerta, M. Mozerov, F. Roca, and J. Gonzalez. Moving cast shadows detection methods for video surveillance applications. *Augmented Vision and Reality*, September 2012.

[2] M. Anderson, G. Ballard, J. Demme, and K. Keutzer. Communication-avoiding QR decomposition for GPUs. *Technical Report, ECCS,*, 2010.

[3] M. Anderson, G. Ballard, J. Demme, and K. Keutzer. Communication-avoiding QR decomposition for GPUs. *IEEE International Parallel and Distributed Processing Symposium, IPDPS 2011*, 2011.

[9] http://soi.stanford.edu/rmunk/PROPACK/

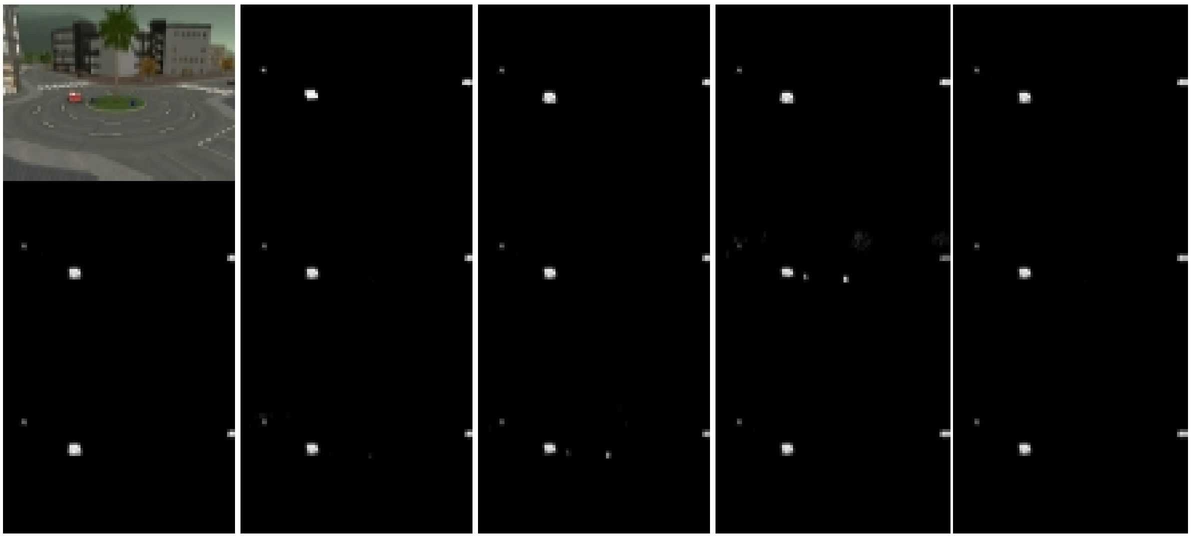

Figure 2: Sequence Rotary 122 (Frame 1250) - First Row: Original image, GT, EALM [31], IALM [31], ADM [70], Deuxime ligne : LADMAP [33], LSADM [15], LDAM [50], BLWS [34], NSA [4], TFOCS [7], Third row : BRPCA [13], VBRPCA [6], GoDec [71], SemiSoft GoDec [71].

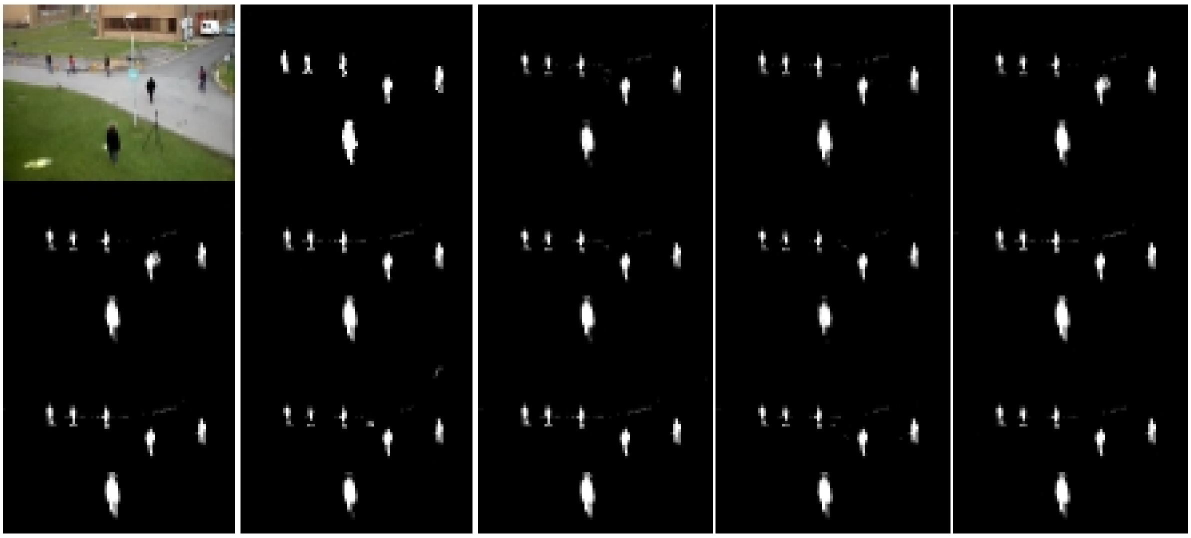

Figure 3: Sequence Video ”Wandering student” (Frame 00000651)- Premire ligne : Image originale, GT, EALM [31], IALM [31], ADM [70], Deuxime ligne : LADMAP [33], LSADM [15], LDAM [50], BLWS [34], NSA [4], TFOCS [7], Troisime ligne : BRPCA [13], VBRPCA [6], GoDec [71], SemiSoft GoDec [71].

Methods	EALM [31]		IALM [31]		ADM (LRSD) [70]		LADMAP [33]		LSADM [15]		LADM (LMaFit) [50]	
	SVDs	CPU	SVDs	CPU	SVDs	CPU	SVDs	CPU	SVDs	CPU	SVDs	CPU
Sequence	550	00:40:15	38	00:03:47	510	00:35:20	16	00:05:20	43	00:04:03	35	00:04:55

Table 4: Time performance of RPCA algorithms: PCP.

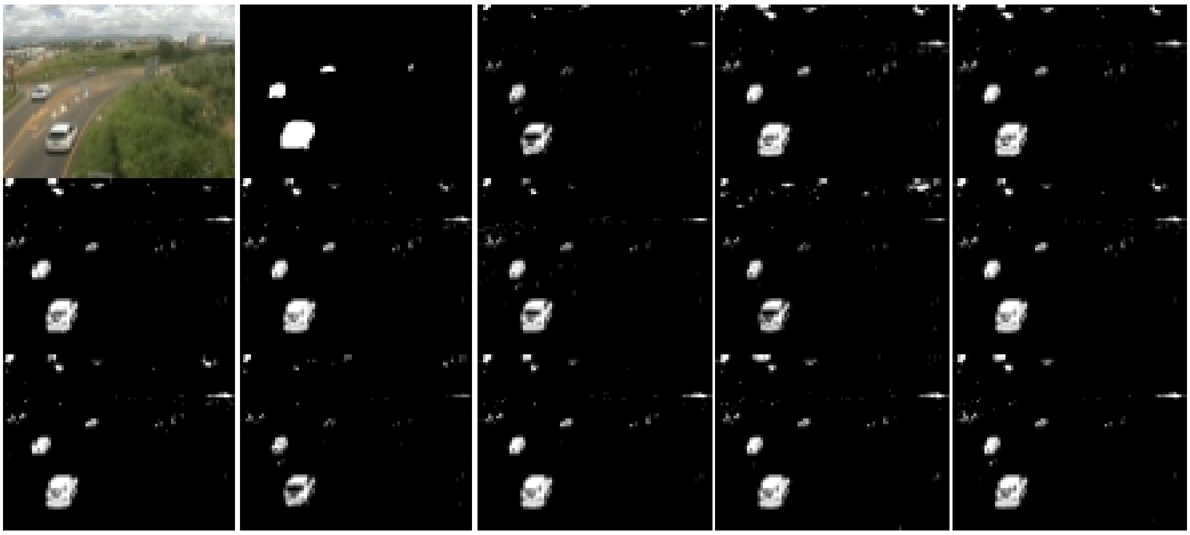

Figure 4: Sequence Video ”Traffic during windy day” (Frame 00000140) - Premire ligne : Image originale, GT, EALM [31], IALM [31], ADM [70], Deuxime ligne : LADMAP [33], LSADM [15], LDAM [50], BLWS [34], NSA [4], TFOCS [7], Troisime ligne : BRPCA [13], VBRPCA [6], GoDec [71], SemiSoft GoDec [71].

	RPCA-SPCP						Bayesian RPCA				Approximated RPCA			
Methods	ASALM [55]		NSA [4]		PSPG [5]		BRPCA [13]		VBRPCA [6]		GoDec [71]		Semi Soft GoDec [71]	
	SVDs	CPU	SVDs	CPU	SVDs	CPU	SVDs	CPU	SVDs	CPU	SVDs	CPU	SVDs	CPU
Sequence	94	00:15:17	19	00:03:07	23	00:01:05	-	00:04:01	-	00:01:07	-	00:00:50	-	00:00:55

Table 5: Time performance of RPCA algorithms: SPCP, Bayesian RPCA and Approximated RPCA.

Method	Algorithms	Measure	Street					Rotary					Average
			112	212	312	412	512	122	222	322	422	522	
PCA	SVD	Recall	0.833	0.873	0.850	0.876	0.876	0.866	0.885	0.808	0.834	0.863	-
Oliver et al.[39]		Precision	0.960	0.944	0.966	0.945	0.898	0.895	0.910	0.946	0.775	0.869	-
		F-measure	0.892	0.907	0.805	0.809	0.787	0.880	0.897	0.871	0.803	0.866	0.851
RPCA-PCP	EALM	Recall	0.757	0.749	0.683	0.666	0.659	0.806	0.801	0.719	0.618	0.746	-
Candes et al. [10]	Lin et al. [31]	Precision	0.981	0.971	0.972	0.950	0.756	0.905	0.903	0.910	0.912	0.789	-
		F-measure	0.855	0.846	0.802	0.783	0.704	0.853	0.849	0.803	0.737	0.767	0.799
	IALM	Recall	0.924	0.839	0.891	0.888	0.827	0.893	0.900	0.891	0.855	0.855	-
	Lin et al. [31]	Precision	0.812	0.961	0.869	0.893	0.814	0.929	0.919	0.890	0.922	0.897	-
		F-measure	0.865	0.896	0.880	0.891	0.820	**0.911**	0.909	0.890	**0.887**	0.875	0.882
	ADM	Recall	0.849	0.841	0.863	0.887	0.830	0.877	0.878	0.861	0.808	0.868	-
	(LRSD)Yuan and Yang [70]	Precision	0.945	0.965	0.900	0.898	0.822	0.937	0.938	0.894	0.941	0.886	-
		F-measure	0.894	0.899	0.881	0.892	0.826	0.906	0.907	0.877	0.869	0.877	0.882
	LADMAP	Recall	0.849	0.841	0.874	0.888	0.831	0.877	0.880	0.863	0.805	0.868	-
	Lin et al. [33]	Precision	0.945	0.965	0.891	0.898	0.823	0.937	0.937	0.895	0.943	0.887	-
		F-measure	0.894	0.899	0.883	0.893	0.827	0.906	0.907	0.879	0.868	0.827	0.878
	LSADM	Recall	0.874	0.857	0.906	0.862	0.840	0.878	0.880	0.892	0.782	0.830	-
	Goldfarb et al. [15]	Precision	0.830	0.965	0.867	0.935	0.742	0.940	0.938	0.892	0.956	0.869	-
		F-measure	**0.912**	**0.908**	**0.886**	**0.897**	0.788	0.908	0.908	**0.892**	0.860	0.849	0.880
	LADM	Recall	0.829	0.803	0.777	0.720	0.740	0.875	0.862	0.750	0.747	0.792	-
	(LMaFit) Shen et al. [50]	Precision	0.979	0.984	0.963	0.966	0.724	0.944	0.953	0.902	0.910	0.832	-
		F-measure	0.898	0.884	0.860	0.825	0.732	0.908	0.906	0.819	0.821	0.812	0.846
	BLWS	Recall	0.700	0.786	0.697	0.785	0.664	0.663	0.633	0.603	0.620	0.793	-
	Lin and Wei [34]	Precision	0.981	0.847	0.971	0.848	0.966	0.921	0.560	0.538	0.775	0.711	-
		F-measure	0.817	0.816	0.812	0.815	0.787	0.771	0.594	0.569	0.689	0.750	0.742
RPCA-SPCP	NSA	Recall	0.857	0.856	0.872	0.850	0.838	0.867	0.871	0.892	0.786	0.837	-
Zhou et al. [72]	Aybat et al. [4]	Precision	0.969	0.966	0.898	0.946	0.747	0.949	0.945	0.891	0.948	0.861	-
		F-measure	0.910	0.907	0.885	0.896	0.790	0.906	0.906	0.891	0.860	0.849	0.880
RPCA-QPCP	TFOCS	Recall	0.910	0.843	0.867	0.903	0.834	0.898	0.892	0.892	0.831	0.877	-
Becker et al. [7]	Becker et al. [7]	Precision	0.830	0.965	0.899	0.889	0.824	0.924	0.932	0.887	0.940	0.879	-
		F-measure	0.868	0.900	0.882	0.896	**0.829**	**0.911**	**0.912**	0.889	0.882	**0.878**	**0.885**
RPCA-BPCP	ALM	Recall	0.757	0.806	0.749	0.801	0.683	0.719	0.666	0.618	0.659	0.746	-
Tang and Nehorai [54]	Tang and Nehorai [54]	Precision	0.981	0.905	0.971	0.903	0.972	0.910	0.950	0.912	0.756	0.789	-
		F-Measure	0.855	0.853	0.846	0.849	0.802	0.803	0.783	0.737	0.704	0.767	0.799
Bayesian RPCA	Bayesian RPCA	Recall	0.809	0.874	0.776	0.666	0.658	0.806	0.661	0.679	0.624	0.746	-
	Ding et al. [13]	Precision	0.978	0.930	0.975	0.950	0.934	0.905	0.889	0.660	0.743	0.789	-
		F-measure	0.885	0.901	0.864	0.783	0.772	0.853	0.758	0.669	0.679	0.767	0.793
	Variational BRPCA	Recall	0.825	0.844	0.788	0.847	0.859	0.820	0.768	0.761	0.803	0.792	-
	Babacan et al. [6]	Precision	0.983	0.925	0.987	0.921	0.907	0.897	0.961	0.907	0.719	0.778	-
		F-measure	0.897	0.882	0.876	0.883	0.882	0.857	0.853	0.828	0.759	0.785	0.850
Approximated RPCA	GoDec	Recall	0.840	0.873	0.921	0.870	0.900	0.857	0.829	0.762	0.778	0.827	-
	Zhou and Tao [71]	Precision	0.966	0.941	0.874	0.944	0.852	0.897	0.865	0.957	0.686	0.831	-
		F-measure	0.899	0.906	0.897	0.906	0.875	0.877	0.846	0.848	0.729	0.829	0.861
	SemiSoft GoDec	Recall	0.841	0.875	0.850	0.868	0.866	0.822	0.879	0.792	0.813	0.866	-
	Zhou and Tao [71]	Precision	0.965	0.942	0.968	0.948	0.902	0.900	0.921	0.953	0.750	0.837	-
		F-measure	0.899	0.907	**0.905**	**0.906**	**0.884**	0.859	0.900	0.865	0.781	0.851	0.876
Adaptive MOG		Recall	0.927	0.927	0.897	0.861	0.921	0.923	0.931	0.897	0.843	0.934	-
Shimada et al. [51]		Precision	0.866	0.868	0.580	0.526	0.619	0.886	0.890	0.626	0.535	0.840	-
		F-measure	0.896	0.896	0.705	0.653	0.740	0.904	0.910	0.738	0.655	**0.884**	0.798

Table 6: Evaluation results using the synthetic videos for evaluation phase.

Method	Algorithms	Measure	Real Videos									Average
			001	002	003	004	005	006	007	008	009	
PCA Oliver et al. [39]	SVD	Recall	0.818	0.672	0.890	0.824	0.799	0.793	0.580	0.753	0.861	-
		Precision	0.740	0.861	0.953	0.833	0.635	0.783	0.826	0.785	0.953	-
		F-measure	0.777	0.755	0.920	0.829	0.700	0.788	0.682	0.768	0.905	0.791
RPCA-PCP Candes et al. [10]	EALM Lin et al. [31]	Recall	0.723	0.765	0.878	0.839	0.660	0.800	0.739	0.670	0.748	-
		Precision	0.554	0.855	0.958	0.844	0.543	0.729	0.899	0.728	0.965	-
		F-measure	0.628	**0.807**	0.916	0.842	0.596	0.763	0.811	0.698	0.843	0.697
	IALM Lin et al. [31]	Recall	0.847	0.664	0.909	0.840	0.785	0.792	0.583	0.767	0.856	-
		Precision	0.735	0.873	0.947	0.827	0.632	0.793	0.833	0.781	0.957	-
		F-measure	0.787	0.754	0.928	0.833	0.701	0.792	0.686	0.774	0.904	0.794
	ADM (LRSD)Yuan and Yang [70]	Recall	0.840	0.653	0.886	0.831	0.780	0.790	0.577	0.758	0.863	-
		Precision	0.751	0.876	0.946	0.820	0.633	0.782	0.828	0.781	0.952	-
		F-measure	**0.793**	0.749	0.915	0.825	0.699	0.786	0.680	0.769	**0.906**	0.791
	LADMAP Lin et al. [33]	Recall	0.840	0.671	0.887	0.831	0.774	0.794	0.573	0.754	0.863	-
		Precision	0.751	0.865	0.945	0.818	0.634	0.782	0.840	0.791	0.952	-
		F-measure	**0.793**	0.756	0.915	0.824	0.697	0.788	0.681	0.772	**0.906**	0.792
	LSADM Goldfarb et al. [15]	Recall	0.842	0.685	0.933	0.870	0.793	0.805	0.599	0.770	0.850	-
		Precision	0.661	0.873	0.952	0.879	0.624	0.805	0.842	0.783	0.959	-
		F-measure	0.741	0.768	**0.942**	0.875	0.699	**0.805**	0.700	0.776	0.901	0.800
	LADM (LMaFit) Shen et al. [50]	Recall	0.788	0.671	0.901	0.834	0.747	0.803	0.581	0.751	0.770	-
		Precision	0.594	0.837	0.962	0.872	0.588	0.770	0.818	0.781	0.971	-
		F-measure	0.678	0.745	0.931	0.852	0.658	0.786	0.679	0.766	0.859	0.772
	BLWS Lin and Wei [34]	Recall	0.726	0.767	0.842	0.811	0.688	0.805	0.752	0.676	0.705	-
		Precision	0.549	0.846	0.963	0.835	0.528	0.714	0.898	0.635	0.959	-
		F-measure	0.625	0.805	0.899	0.823	0.598	0.757	0.818	0.655	0.812	**0.839**
RPCA-SPCP Zhou et al. [72]	NSA Aybat et al. [4]	Recall	0.838	0.765	0.934	0.875	0.660	0.805	0.599	0.771	0.748	-
		Precision	0.664	0.855	0.951	0.877	0.543	0.805	0.843	0.781	0.965	-
		F-measure	0.741	**0.807**	**0.942**	**0.876**	0.596	**0.805**	0.700	0.776	0.843	0.787
RPCA-QPCP Becker et al. [7]	TFOCS Becker et al. [7]	Recall	0.840	0.655	0.900	0.843	0.793	0.795	0.585	0.763	0.856	-
		Precision	0.736	0.879	0.948	0.823	0.633	0.792	0.833	0.795	0.956	-
		F-measure	0.785	0.751	0.923	0.833	**0.704**	0.793	0.687	**0.779**	0.904	0.794
RPCA-BPCP Tang and Nehorai [54]	ALM Tang and Nehorai [54]	Recall	0.723	0.765	0.878	0.839	0.660	0.800	0.749	0.670	0.748	-
		Precision	0.554	0.855	0.958	0.844	0.543	0.729	0.900	0.728	0.965	-
		F-measure	0.628	**0.807**	0.916	0.842	0.596	0.763	**0.817**	0.698	0.843	0.767
Bayesian RPCA	BRPCA Ding et al. [13]	Recall	0.728	0.774	0.886	0.838	0.695	0.792	0.593	0.662	0.740	-
		Precision	0.553	0.857	0.949	0.824	0.536	0.732	0.839	0.732	0.966	-
		F-measure	0.629	**0.813**	0.917	0.831	0.605	0.761	0.695	0.695	0.839	0.753
	Variational BRPCA Babacan et al. [6]	Recall	0.834	0.690	0.935	0.875	0.781	0.817	0.606	0.755	0.847	-
		Precision	0.621	0.862	0.949	0.874	0.614	0.809	0.843	0.781	0.950	-
		F-measure	0.712	0.766	**0.942**	0.874	0.687	**0.813**	0.705	0.768	0.896	0.795
Approximated RPCA	GoDec Zhou and Tao [71]	Recall	0.833	0.701	0.910	0.858	0.771	0.820	0.615	0.747	0.850	-
		Precision	0.594	0.831	0.957	0.877	0.612	0.786	0.775	0.751	0.896	-
		F-measure	0.693	0.761	0.933	0.868	0.682	0.802	0.686	0.749	0.872	0.782
	SemiSoft GoDec Zhou and Tao [71]	Recall	0.816	0.641	0.918	0.831	0.785	0.793	0.588	0.743	0.832	-
		Precision	0.697	0.855	0.959	0.843	0.638	0.781	0.792	0.778	0.965	-
		F-measure	0.752	0.733	0.938	0.837	**0.704**	0.787	0.675	0.760	0.894	0.787
Adaptive MOG Shimada et al. [51]		Recall	0.949	0.680	0.959	0.929	0.854	0.880	0.791	0.823	0.928	-
		Precision	0.782	0.646	0.880	0.680	0.535	0.736	0.703	0.595	0.890	-
		F-measure	**0.857**	0.662	0.918	0.785	0.658	0.802	0.744	0.691	**0.909**	0.780

Table 7: Evaluation results using the real videos for evaluation phase.

	PCA [39]	RPCA-PCP [15]	RPCA-SPCP [72]	RPCA-QPCP [7]	RPCA-BPCP [54]	Bayesian RPCA [6]	Approximated RPCA [71]
Synthetic videos	0.851	0.880	0.880	**0.885**	0.799	0.850	0.876
Real videos	0.791	**0.800**	0.787	0.794	0.767	0.795	0.787

Table 8: F-Measure for the synthetic and real videos.

[4] N. Aybat, D. Goldfarb, and G. Iyengar. Fast first-order methods for stable principal component pursuit. *Preprint*, 2011.

[5] N. Aybat, D. Goldfarb, and G. Iyengar. Efficient algorithms for robust and stable principal component pursuit. *Preprint*, November 2012.

[6] S. Babacan, M. Luessi, R. Molina, and A. Katsaggelos. Sparse bayesian methods for low-rank matrix estimation. *IEEE Transactions on Signal Processing*, 2011.

[7] S. Becker, E. Candes, and M. Grant. TFOCS: flexible first-order methods for rank minimization. *Low-rank Matrix Optimization Symposium, SIAM Conference on Optimization*, 2011.

[8] T. Bouwmans. Recent advanced statistical background modeling for foreground detection: A systematic survey. *RPCS*, 4(3):147–176, November 2011.

[9] J. Cai, E. Candes, and Z. Shen. A singular value thresholding algorithm for matrix completion. *International Journal of ACM*, May 2008.

[10] E. Candes, X. Li, Y. Ma, and J. Wright. Robust principal component analysis? *International Journal of ACM*, 58(3), May 2011.

[11] V. Chandrasekharan, S. Sanghavi, P. Parillo, and A. Wilsky. Rank-sparsity incoherence for matrix decomposition. *Preprint*, 2009.

[12] R. Chartrand. Non convex splitting for regularized low-rank and sparse decomposition. *IEEE Transactions on Signal Processing*, 2012.

[13] X. Ding, L. He, and L. Carin. Bayesian robust principal component analysis. *IEEE Transaction on Image Processing*, 2011.

[14] M. Kleinsteuber F. Seidel, C. Hage. pROST : A smoothed l_p-norm robust online subspace tracking method for realtime background subtraction in video. *Machine Vision and Applications, Special issue on background modeling for foreground detection in real-world dynamic scenes*, December 2013.

[15] D. Goldfarb, S. Ma, and K. Scheinberg. Fast alternating linearization methods for minimizing the sum of two convex function. *Preprint, Mathematical Programming Series A*, 2010.

[16] T. Goldstein and S. Osher. The split bregman method for l_1-regularized problems. *SIAM Journal of Image Science*, 2(2):323–343, 2009.

[17] C. Guyon, T. Bouwmans, and E. Zahzah. Foreground detection based on low-rank and block-sparse matrix decomposition. *IEEE International Conference on Image Processing, ICIP 2012*, September 2012.

[18] C. Guyon, T. Bouwmans, and E. Zahzah. Foreground detection by robust PCA solved via a linearized alternating direction method. *International Conference on Image Analysis and Recognition, ICIAR 2012*, June 2012.

[19] C. Guyon, T. Bouwmans, and E. Zahzah. Foreground detection via robust low rank matrix decomposition including spatio-temporal constraint. *International Workshop on Background Model Challenges, ACCV 2012*, November 2012.

[20] C. Guyon, T. Bouwmans, and E. Zahzah. Foreground detection via robust low rank matrix factorization including spatial constraint with iterative reweighted regression. *International Conference on Pattern Recognition, ICPR 2012*, November 2012.

[21] C. Guyon, T. Bouwmans, and E. Zahzah. Moving object detection via robust low rank matrix decomposition with IRLS scheme. *International Symposium on Visual Computing, ISVC 2012*, pages 665–674, July 2012.

[22] C. Guyon, T. Bouwmans, and E. Zahzah. Robust principal component analysis for background subtraction: Systematic evaluation and comparative analysis. *INTECH, Principal Component Analysis, Book 1, Chapter 12*, pages 223–238, March 2012.

[23] C. Hage and M. Kleinsteuber. Robust PCA and subspace tracking from incomplete observations using l_0-surrogates. *Preprint*, 2012.

[24] T. Hastie, R. Tibshirani, and J. Friedman. The elements of statistical learning: Data mining, inference and prediction. *2nd Edition Springer*, February 2009.

[25] J. He, L. Balzano, and J. Luiz. Online robust subspace tracking from partial information. *IT 2011*, September 2011.

[26] J. He, L. Balzano, and A. Szlam. Incremental gradient on the grassmannian for online foreground and background separation in subsampled video. *International on Conference on Computer Vision and Pattern Recognition, CVPR 2012*, June 2012.

[27] H. Jiang, W. Deng, and Z. Shen. Surveillance video processing using compressive sensing. *Inverse Problems and Imaging*, 6(2):201–214, 2012.

[28] S. Kumar, M. Mohri, and A. Talwalkar. Ensemble Nystrom method. *NIPS*, 2009.

[29] L. Li and W. Huang. Statistical modeling of complex background for foreground object detection. *IEEE Transaction on Image Processing*, 13(11):1459–1472, November 2004.

[30] L. Li, W. Huang, I. Gu, and Q. Tian. Statistical modeling of complex backgrounds for foreground object detection. *IEEE Transaction on Image Processing*, pages 1459–1472, 2004.

[31] Z. Lin, M. Chen, L. Wu, and Y. Ma. The augmented Lagrange multiplier method for exact recovery of corrupted low-rank matrices. *UIUC Technical Report*, November 2009.

[32] Z. Lin, A. Ganesh, J. Wright, L. Wu, M. Chen, and Y. Ma. Fast convex optimization algorithms for exact recovery of a corrupted low-rank matrix. *UIUC Technical Report*, August 2009.

[33] Z. Lin, R. Liu, and Z. Su. Linearized alternating direction method with adaptive penalty for low-rank representation. *NIPS 2011*, December 2011.

[34] Z. Lin and S. Wei. A block lanczos with warm start technique for accelerating nuclear norm minimization algorithms. *Preprint*, 2010.

[35] R. Liu, Z. Lin, S. Wei, and Z. Su. Solving principal component pursuit in linear time via l_1 filtering. *International Journal on Computer Vision, IJCV 2011*, 2011.

[36] S. Ma. Algorithms for sparse and low-rank optimization: Convergence, complexity and applications. *Thesis*, June 2011.

[37] L. Mackey, A. Talwalkar, and M. Jordan. Divide-and-conquer matrix factorization. *Neural Information Processing Systems, Neural Information Processing Systems, NIPS 2011*, December 2011.

[38] Y. Mu, J. Dong, X. Yuan, and S. Yan. Accelerated low-rank visual recovery by random projection. *International Conference on Computer Vision, CVPR 2011*, 2011.

[39] N. Oliver, B. Rosario, and A. Pentland. A Bayesian computer vision system for modeling human interactions. *ICVS 1999*, January 1999.

[40] G. Pope, M. Baumann, C. Studery, and G. Durisi. Real-time principal component pursuit. *Asilomar Conference on Signals, Systems, Computation*, November 2011.

[41] C. Qiu and N. Vaswani. Real-time robust principal components pursuit. *International Conference on Communication Control and Computing*, 2010.

[42] C. Qiu and N. Vaswani. ReProCS: a missing link between recursive robust PCA and recursive sparse recovery in large but correlated noise. *Preprint*, 2011.

[43] C. Qiu and N. Vaswani. Support predicted modified-CS for recursive robust principal components' pursuit. *IEEE International Symposium on Information Theory, ISIT 2011*, 2011.

[44] C. Qiu and N. Vaswani. Automated recursive projected CS (ReProCS) for real-time video layering. *International Conference on Computer Vision and Pattern Recognition, CVPR 2012,*, 2012.

[45] I. Ramirez and G. Sapiro. Low-rank data modeling via the min-

imum description length principle. *ICASSP 2012*, 2012.

[46] I. Ramirez and G. Sapiro. An MDL framework for sparse coding and dictionary learning. *Preprint*, 2012.

[47] P. Rodriguez and B. Wohlberg. Fast principal component pursuit via alternating minimization. *IEEE International Conference on Image Processing, ICIP 2013*, September 2013.

[48] M. Shah, J. Deng, and B. Woodford. Video background modeling: Recent approaches, issues and our solutions. *Machine Vision and Applications, Special issue on background modeling for foreground detection in real-world dynamic scenes*, December 2013.

[49] Y. Sheikh and M. Shah. Bayesian modeling of dynamic scenes for object detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27:1778–1792, 2005.

[50] Y. Shen, Z. Wen, and Y. Zhang. Augmented Lagrangian alternating direction method for matrix separation based on low-rank factorization. *Preprint*, January 2011.

[51] A. Shimada, D. Arita, and R. Taniguchi. Dynamic control of adpative mixture of Gaussians background model. *IEEE International Conference on Advanced Video and Signal Based Surveillance, AVSS 2006*, 2006.

[52] A. Shimada, Y. Nonaka, H. Nagahara, and R. Taniguchi. Case-based background modeling -towards low-cost and high-performance background model. *Machine Vision and Applications, Special issue on background modeling for foreground detection in real-world dynamic scenes*, December 2013.

[53] C. Stauffer and E. Grimson. Adaptive background mixture models for real-time tracking. *IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, pages 246–252, 1999.

[54] G. Tang and A. Nehorai. Robust principal component analysis based on low-rank and block-sparse matrix decomposition. *CISS 2011*, 2011.

[55] M. Tao and X. Yuan. Recovering low-rank and sparse components of matrices from incomplete and noisy observations. *SIAM Journal on Optimization*, 21(1):57–81, 2011.

[56] F. De La Torre and M. Black. A robust principal component analysis for computer vision. *International Conference on Computer Vision*, 2001.

[57] F. De La Torre and M. Black. A framework for robust subspace learning. *International Journal on Computer Vision*, pages 117–142, 2003.

[58] K. Toyama, J. Krumm, B. Brumitt, and B. Meyers. Wallflower: Principles and practice of background maintenance. *ICCV*, pages 255–261, September 1999.

[59] A. Vacavant, T. Chateau, A. Wilhelm, and L. Lequievre. A benchmark dataset for foreground/background extraction. *International Workshop on Background Models Challenge, ACCV 2012*, November 2012.

[60] Y. Wang, Y. Liu, and L. Wu. Study on background modeling method based on robust principal component analysis. *Annual Conference on Electrical and Control Engineering, ICECE 2011*, pages 6787–6790, September 2011.

[61] A. Waters, A. Sankaranarayanan, and R. Baraniuk. SpaRCS: recovering low-rank and sparse matrices from compressive measurements. *Neural Information Processing Systems, NIPS 2011*, December 2011.

[62] A. Waters, A. Sankaranarayanan, and R. Baraniuk. SpaRCS: recovering low-rank and sparse matrices from compressive measurements. *Technical Report*, 2011.

[63] B. Wohlberg, R. Chartrand, and J. Theiler. Local principal component analysis for nonlinear datasets. *International Conference on Acoustics, Speech, and Signal Processing, ICASSP 2012*, March 2012.

[64] J. Wright, Y. Peng, Y. Ma, A. Ganesh, and S. Rao. Robust principal component analysis: Exact recovery of corrupted low-rank matrices by convex optimization. *Neural Information Processing Systems, NIPS 2009*, December 2009.

[65] X.Liu, Z. Wen, and Y. Zhang. Limited memory block Krylov subspace optimization for computing dominant singular value decompositions. *Preprint*, 2012.

[66] H. Xu, C. Caramanis, and S. Sanghavi. Robust PCA via outlier pursuit. *NIPS 2010*, 2010.

[67] Y. Xue, X. Gu, and X. Cao. Motion saliency detection using low-rank and sparse decomposition. *International Conference on Acoustics, Speech, and Signal Processing,ICASSP 2012*, March 2012.

[68] J. Yang and X. Yuan. Linearized augmented Lagrangian and alternating direction methods for nuclear norm minimization. *Preprint*, 2011.

[69] M. Yang. Background modeling from surveillance video using rank minimization. *Artificial Intelligence and Computational Intelligence, AICI 2012*, pages 769–774, 2012.

[70] X. Yuan and J. Yang. Sparse and low-rank matrix decomposition via alternating direction methods. *Optimization Online*, November 2009.

[71] T. Zhou and D. Tao. GoDec: randomized low-rank and sparse matrix decomposition in noisy case. *International Conference on Machine Learning, ICML 2011*, 2011.

[72] Z. Zhou, X. Li, J. Wright, E. Candes, and Y. Ma. Stable principal component pursuit. *IEEE ISIT Proceedings*, pages 1518–1522, June 2010.