AN INCREMENTAL PRINCIPAL COMPONENT PURSUIT ALGORITHM VIA PROJECTIONS ONTO THE ℓ_1 BALL

Paul Rodríguez[*]

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

Brendt Wohlberg[†]

T-5 Applied Mathematics and Plasma Physics
Los Alamos National Laboratory
Los Alamos, NM 87545, USA

ABSTRACT

Video background modeling, used to detect moving objects in digital videos, is a ubiquitous pre-processing step in computer vision applications. Principal Component Pursuit (PCP) PCP is among the leading methods for this problem. In this paper we proposed a new convex formulation for PCP, substituting the standard ℓ_1 regularization with a projection onto the ℓ_1-ball. This formulation offers an advantage over the known incremental PCP methods in practical parameter selection and ghosting suppression, while retaining the ability to be implemented in a fully incremental fashion, keeping all the desired properties related to such PCP methods (low memory footprint, adaptation to changes in the background, computational complexity that allows online processing).

***Index Terms*—** Incremental Principal Component Pursuit, ℓ_1 Ball projection, Video Background Modeling.

1. INTRODUCTION

Video background modeling, which consists of segmenting the moving objects or "foreground" from the static "background", is an important task in several applications. Principal Component Pursuit (PCP) is currently considered to be the most effective formulation for this problem [1]. PCP was introduced in [2] as the non-convex optimization problem given by (1)

$$\arg\min_{L,S} \quad \mathrm{rank}(L) + \lambda \|S\|_0 \ \text{ s.t. } D = L + S \ , \qquad (1)$$

where $\lambda > 0$ is a fixed and global regularization parameter, $D \in \mathbb{R}^{m \times n}$ is the observed video of n frames, each of size $m = N_r \times N_c \times N_d$ (rows, columns and depth or channels respectively), $L \in \mathbb{R}^{m \times n}$ is a low rank matrix representing the background and $S \in \mathbb{R}^{m \times n}$ is a sparse matrix representing the foreground, i.e. moving objects.

While most PCP algorithms (for a complete list see [1, 3]), including the Augmented Lagrange Multiplier (ALM) and inexact ALM (iALM) algorithms [4, 5] are directly based on the original convex relation (2)

$$\arg\min_{L,S} \|L\|_* + \lambda \|S\|_1 \ \text{ s.t. } D = L + S \ , \qquad (2)$$

this is not the only possible tractable problem that can be derived from (1) [3]. Although theoretical guidance is available for selecting a minimax optimal regularization parameter λ in (2) [6], practical problems do not fully satisfy the idealized assumptions, and thus λ often has to be heuristically tuned.

In this paper we propose the novel[1] decomposition

$$\arg\min_{L,S} \frac{1}{2}\|L+S-D\|_F^2 \ \text{ s.t. } \|S\|_1 \leq \tau, \ \mathrm{rank}(L) \leq r \ , \ (3)$$

which is a convex relaxation of (1). We show that a simple practical scheme can be derived to adaptively select the parameter τ, and that (3) can be easily solved in an incremental fashion (i.e. one frame at a time), and thus the τ parameter can be adaptively estimated for every frame. Interestingly, the τ parameter can also be spatially adapted, dramatically reducing the ghosting effects[2] that are usually observed in incremental methods (see [7] for a recent discussion of this topic).

2. PREVIOUS RELATED WORK

In this section we give a brief overview of other incremental PCP methods, with a particular focus on the incremental PCP algorithm [8] since, like our proposed algorithm (see Section 3), it also uses rank-1 modifications [9] for thin singular value decomposition (SVD) to achieve its computational efficiency. Finally we also describe the ℓ_1-ball projection problem, which is central to our proposed method.

[*]This research was supported by the "Fondo para la Innovación, la Ciencia y la Tecnología" (Fincyt) Program

[†]This research was supported by the NNSA's Laboratory Directed Research and Development Program.

[1]The known decompositions for the PCP problem are listed in [3, Table 4]. The most closely related alternatives are GoDec and DRMF; they both used the ℓ_0 (additionally DRMF also used the mix $\ell_{2,0}$) norm as constraint, resulting in algorithms fundamentally different from ours (see Section 3).

[2]Ghosting occurs when the foreground estimate includes phantoms or smear replicas from actual moving objects, or from objects that really belong to the background.

2.1. Incremental PCP methods

To the best of our knowledge, recursive projected compressive sensing (ReProCS) [10, 11] along with Grassmannian robust adaptive subspace tracking algorithm (GRASTA) [12], ℓ_p-norm robust online subspace tracking (pROST) [13], Grassmannian online subspace updates with structured sparsity (GOSUS) [14] and the incremental PCP (incPCP) [8] are the only PCP-like methods for the video background modeling problem that are considered to be incremental. However, except for incPCP, these methods have a batch initialization/training stage as the default initial background estimate[3].

2.2. Intuitive description of the incPCP algorithm

The incPCP algorithm [15, 8, 16] is a computationally efficient solution to the amFastPCP algorithm [17]. The amFastPCP algorithm proposed the convex relaxation

$$\arg\min_{L,S} \frac{1}{2}\|L+S-D\|_F^2 + \lambda\|S\|_1 \quad \text{s.t. } \operatorname{rank}(L) \le r \quad (4)$$

as an alternative to the original relaxation presented in (2).

In [17], it was shown that (4) can be solved via an alternating optimization (AO) [18] procedure, since it seems natural to split (4) into a low-rank approximation[4], i.e. $\arg\min_L \frac{1}{2}\|L+S-D\|_F^2$ s.t. $\operatorname{rank}(L) \le r$, with fixed S, soft-thresholding[5], i.e. $\arg\min_S \frac{1}{2}\|L+S-D\|_F^2 + \lambda\|S\|_1$, with fixed L computed in the previous iteration. The solution obtained via this AO procedure is of comparable quality to the solution of the original PCP problem [17].

Furthermore, the low-rank approximation sub-problem can be solved via a computationally efficient incremental procedure, based on rank-1 modifications for thin SVD (see [9] and the many references therein) and thus (4) can also be easily solved incrementally, since the soft-thresholding step can be trivially computed in an incremental fashion. The resulting incPCP algorithm [8] is a fully incremental PCP algorithm for video background modeling, obtaining similar results to batch PCP algorithms by processing one frame at a time, and being able to adapt to changes in the background.

2.3. Projection onto the ℓ_1 ball

The ℓ_1-ball projection problem, which is used in several imaging and machine learning problems, is defined as

$$\operatorname{proj}_{\|\cdot\|_1}(\mathbf{u}) = \min_{\mathbf{x}} \frac{1}{2}\|\mathbf{x}-\mathbf{u}\|_2^2 \quad \text{s.t. } \|\mathbf{x}\|_1 \le \tau, \quad (5)$$

where $\mathbf{x}, \mathbf{u} \in \mathbb{R}^m$ and $\tau > 0$.

If $\|\mathbf{u}\|_1 \le \tau$, then $\mathbf{x}^* = \mathbf{u}$ is the solution to (5). When $\|\mathbf{u}\|_1 > \tau$, the optimal solution to (5) must satisfy $\|\mathbf{x}^*\|_1 = \tau$ and it is given by soft-thresholding [19, 20]:

$$\mathbf{x}^* = \operatorname{shrink}(\mathbf{u}, \lambda(\tau)) = \operatorname{sign}(\mathbf{u}) \cdot \max\{0, |\mathbf{u}| - \lambda(\tau)\}, \quad (6)$$

where $\lambda(\tau)$ is a threshold that depends on the parameter τ, and it is usually found by performing a sorting of the elements of $\mathbf{u}$ in decreasing order.

There are several efficient algorithms to solve (5). While [19] and [20] are the most well-known ones ([20] being faster than [19]), for our computational results in Section 4 we use a novel algorithm based on an accelerated Newton's method applied to (5) [21] that does not perform any type of sorting, and that can be easily parallelize in several architectures, including CUDA.

3. PROPOSED ALGORITHM

We start our description assuming that we have computed the solution to (3) up to frame $k-1$, i.e. L_{k-1} (low-rank) and S_{k-1} (sparse), where $L_{k-1} + S_{k-1} = D_{k-1}$ and $D_{k-1} = D(:, 1:k-1)$ and that we know the partial (thin) SVD of $L_{k-1} = U_r \Sigma_r V_r^T$, where $\Sigma_r \in \mathbb{R}^{r\times r}$. This initialization can be trivially computed when $k = 2$.

If we were to solve (3) from scratch when the next frame $\mathbf{d}_k$ is available, then we would need to minimize

$$\frac{1}{2}\|L_k + S_k - D_k\|_F^2 \ \text{ s.t. } \|S_k\| \le \tau,\ \operatorname{rank}(L_k) \le r, \quad (7)$$

which can be iteratively done via the AO

$$L_k^{(j+1)} = \arg\min_L \|L_k + S_k^{(j)} - D_k\|_F^2 \text{ s.t. } \operatorname{rank}(L_k) \le r \quad (8)$$

$$S_k^{(j+1)} = \arg\min_S \|L_k^{(j+1)} + S_k - D_k\|_F^2 \ \text{ s.t. } \|S_k\|_1 \le \tau \ , (9)$$

where $L_k = [L_{k-1}\, \mathbf{l}_k]$, $S_k = [S_{k-1}\, \mathbf{s}_k]$ and $D_k = [D_{k-1}\, \mathbf{d}_k]$. When $j = 0$, the minimizer of (8) is given by

$$L_k^{(1)} = \operatorname{partialSVD}(D_k - S_k^{(0)}) \ . \quad (10)$$

Since $D_k - S_k^{(0)} = [D_{k-1}\text{-}S_{k-1}\ \mathbf{d}_k] = [L_{k-1}\ \mathbf{d}_k]$, and we know that $L_{k-1} = U_r \Sigma_r V_r^T$, then (10) can be computed via the incremental thin SVD [9] procedure.

The minimizer of (9) is the projection onto the ℓ_1 ball (see (5)) and it would be only applied to the current estimate (since S_{k-1} is known) i.e. $\mathbf{s}_k^{(1)} = \operatorname{proj}_{\|\cdot\|_1}(\mathbf{d}_k - \mathbf{l}_k^{(1)})$, where $\mathbf{l}_k^{(1)}$ is the last column of the current estimate $L_k^{(1)}$. In the next inner loop ($j = 1$) for solving (8) we have a similar situation to (10), i.e. $L_k^{(2)} = \operatorname{partialSVD}(D_k - S_k^{(1)})$, and noting that $D_k - S_k^{(1)} = [D_{k-1} - S_{k-1}\ \ \mathbf{d}_k - \mathbf{s}_k^{(1)}]$, then the solution can be effectively computed using the thin SVD replace [9]

[3] GRASTA and GOSUS can perform the initial background estimation in a non-batch fashion, however the resulting performance is not as good as when the default batch procedure is used; see [8, Section 6]. pROST is closely related to GRASTA, and it shares the same restrictions. All variants of ReProCS also use a batch initialization stage.

[4] The solution is given by $L = U\Sigma_r V'$, where $[U, \Sigma, V] = \operatorname{SVD}(D - S)$ and Σ_r keeps the r largest singular values of the diagonal matrix Σ.

[5] The solution is given by (6), with $\lambda(\tau) = \lambda$.

procedure, since in the previous step we have computed the partial SVD for $[D_{k-1}\text{-}S_{k-1}\ \mathbf{d}_k]$.

The procedure described above is very similar to the incPCP algorithm [8]. The key difference is that in the present case we solve (9), whereas [8] solves

$$S_k^{(j+1)} = \arg\min_{S} \|L_k^{(j+1)} + S_k - D_k\|_F^2 + \lambda\|S_k\|_1. \quad (11)$$

For instance, given an oracle that provides the ideal solution to (8) at frame k, i.e. $\mathbf{l}_k^*$, it is not clear how to choose λ for (11). However, for (9) it is straightforward to set $\tau_k^* = \|\mathbf{d}_k - \mathbf{l}_k^*\|_1$, which amounts to adaptively estimate the best threshold at every frame. Since the solution to (8) is not ideal, we use (12). In our experimental results (see Section 4), we use $\alpha \in [0.5,\ 0.75]$.

$$\tau_k \quad = \quad \alpha \cdot \|\mathbf{d}_k - \mathbf{l}_k\|_1. \quad (12)$$

Furthermore, τ_k can be spatially adapted: given the difference between the current and past sparse approximations, i.e. $\mathbf{z}_k = \mathbf{s}_k - \mathbf{s}_{k-1}$, it is expected that, in the ideal case, $\mathbf{z}_k$ must be zero except in the boundaries of the moving objects, but, due to ghosting effects, this is not the case in practice.

Based on $|\mathbf{z}_k|$, a binary mask $\mathbf{m}_k$ can be found that is set to one in the areas with high probability of being affected by ghosting, and thus

$$\mathbf{s}_k \quad = \quad (\mathbf{1} - \mathbf{m}_k) \cdot \hat{\mathbf{s}}_k + \mathbf{m}_k \cdot \check{\mathbf{s}}_k, \quad (13)$$

where $\hat{\mathbf{s}}_k = \text{proj}_{\|\cdot\|_1}(\mathbf{d}_k - \mathbf{l}_k)$, $\check{\mathbf{s}}_k = \text{proj}_{\|\cdot\|_1}(\mathbf{m}_k \cdot \hat{\mathbf{s}}_k)$ and $\tau_k^{(g)} = \beta \cdot \|\mathbf{m}_k \cdot \hat{\mathbf{s}}_k\|_1$. In our experimental results we use $\beta \in [0.1,\ 0.3]$.

4. COMPUTATIONAL RESULTS

In Table 1 we present F-measure based accuracy results for two challenging videos[6] from the CDnet dataset [23]. The F-measure, which makes use of a binary ground-truth, is defined in (14), where P and R stands for precision and recall respectively, and TP, FN and FP are the number of true positive, false negative and false positive pixels, respectively:

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

In order to compute the F-measure for the proposed algorithm, called ℓ_1B-PCP, as well as for incPCP, GRASTA and GOSUS, a threshold is needed to compute the binary foreground mask. For this purpose we use an automatic segmentation [24] that adapts its threshold for each sparse representation and ensures that all algorithms are fairly treated.

[6]V320: 320×240 pixel, 1700-frame color video sequence, with 1230 ground-truth frames, from a highway camera with lots of cars passing by; V720: 720×576 pixel, 1200-frame color video sequence, 900 ground-truth frames, of a train station with lots of people walking around. Results for other datasets are not included here due to space limitations, however they can be found in [22] along with the corresponding Matlab code.

All simulations presented here were run on an Intel i7-4710HQ (2.5 GHz, 6MB Cache, 32GB RAM) based laptop. The GRASTA[7] and GOSUS[7] implementations are Matlab based with MEX interfaces, while the Matlab implementation of incPCP and ℓ_1B-PCP [22] takes advantage of GPU-enabled (CUDA) Matlab functions (mainly linear algebra). Our ℓ_1B-PCP algorithm uses $\alpha = 0.75$ and $\beta = 0.3$ (see Section 3) for the considered videos.

Video	F-measure / (f.p.s.)					
	grayscale			color		
	incPCP	ℓ_1B-PCP	GRASTA	incPCP	ℓ_1B-PCP	GOSUS
V320	0.745	**0.799**	0.773	0.794	**0.832**	0.549
	(76.9)	(23.9)	(29.4)	(58.8)	(21.2)	(0.3)
V720	0.687	**0.741**	0.169	0.728	**0.784**	0.426
	(31.2)	(6.1)	(4.0)	(30.3)	(4.1)	(0.02)

Table 1. Accuracy performance via the F-measure on the CDnet dataset for the ℓ_1B-PCP, incPCP, GRASTA and GOSUS algorithms. V320 is a $320 \times 240 \times 1700$ video from a highway camera, and V720 is a $720 \times 576 \times 1200$ video from a train station (sizes are indicated in the form rows$\times$columns$\times$frames). The bold values are the largest F-measure values (grayscale and color are treated independently). The inverse of the average processing time per frame, i.e. the average number of frame per seconds (f.p.s.) is also shown for each case.

The accuracy and inverse of the average processing time per frame, i.e. average frame per seconds (f.p.s.) results for the CDnet dataset, listed in Table 1, show that the ℓ_1B-PCP gives superior performance when compared to the considered alternatives. Moreover, ℓ_1B-PCP is equally effective for videos with a moving background, such as in the case of "V320", where waving trees/leaves are observed or fully static background, such as in the case of "V720".

5. CONCLUSIONS

We have presented an incremental PCP algorithm that uses a novel convex formulation for the PCP problem, which instead of using the standard ℓ_1 regularization term for the sparse component, uses a projection onto ℓ_1-ball as a restriction.

Our algorithm, called ℓ_1B-PCP, offers an advantage over the known incremental PCP methods in practical parameter selection and ghosting suppression as shown in our computational results, where the ℓ_1B-PCP achieves the best reconstruction metrics when compared to other leading incremental PCP methods (incPCP, GRASTA and GOSUS). Future work will focus on improving the computational performance of the ℓ_1B-PCP algorithm, which, although two orders of magnitude faster than GOSUS and competitive with GRASTA, is

[7]While the GRASTA and GOSUS algorithms are able to process both grayscale and color video, the GRASTA software implementation [12] can only process grayscale video, and the GOSUS implementation [14] can only process color video, which places some restrictions on our comparisons.

currently $[2 \sim 5]$ times slower than incPCP, the fastest incremental PCP algorithm.

6. REFERENCES

[1] 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*, vol. 122, pp. 22–34, 2014.

[2] 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 *Adv. in Neural Inf. Proc. Sys. (NIPS) 22*, 2009, pp. 2080–2088.

[3] 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*, vol. 23, pp. 1–71, 2017.

[4] G. Liu, Z. Lin, and Y. Yu, "Robust subspace segmentation by low-rank representation.," in *ACM ICML*, 2010, pp. 663–670.

[5] Z. Lin, M. Chen, and Y. Ma, "The augmented lagrange multiplier method for exact recovery of corrupted low-rank matrices," arXiv:1009.5055v2, 2011.

[6] E. Candès, X. Li, Y. Ma, and J. Wright, "Robust principal component analysis?," *J. ACM*, vol. 58, no. 3, May 2011.

[7] P. Rodríguez and B. Wohlberg, "Ghosting suppression for incremental principal component pursuit algorithms," in *IEEE GlobalSIP*, Dec. 2016, pp. 197–201.

[8] P. Rodriguez and B. Wohlberg, "Incremental principal component pursuit for video background modeling," *J. of Math. Imag. and Vis.*, vol. 55, no. 1, pp. 1–18, 2016.

[9] M. Brand, "Fast low-rank modifications of the thin singular value decomposition," *Linear Algebra and its Applications*, vol. 415, no. 1, pp. 20 – 30, 2006.

[10] C. Qiu and N. Vaswani, "Support predicted modified-CS for recursive robust principal components pursuit," in *IEEE ISIT*, 2011, pp. 668–672.

[11] H. Guo, C. Qiu, and N. Vaswani, "An online algorithm for separating sparse and low-dimensional signal sequences from their sum," *IEEE TSP*, vol. 62, no. 16, pp. 4284–4297, Aug 2014.

[12] J. He, L. Balzano, and A. Szlam, "Incremental gradient on the Grassmannian for online foreground and background separation in subsampled video," in *IEEE CVPR*, June 2012, pp. 1568–1575.

[13] F. Seidel, C. Hage, and M. Kleinsteuber, "pROST: a smoothed lp-norm robust online subspace tracking method for background subtraction in video," *Machine Vis. and Apps.*, vol. 25, no. 5, pp. 1227–1240, 2014.

[14] Jia Xu, Vamsi K. Ithapu, Lopamudra Mukherjee, James M. Rehg, and Vikas Singh, "GOSUS: Grassmannian online subspace updates with structured-sparsity," in *IEEE ICCV*, Dec. 2013, pp. 3376–3383.

[15] P. Rodríguez and B. Wohlberg, "A Matlab implementation of a fast incremental principal component pursuit algorithm for video background modeling," in *IEEE ICIP*, Oct. 2014, pp. 3414–3416.

[16] P. Rodriguez and B. Wohlberg, "Translational and rotational jitter invariant incremental principal component pursuit for video background modeling," in *IEEE ICIP*, Sept. 2015, pp. 537–541.

[17] P. Rodríguez and B. Wohlberg, "Fast principal component pursuit via alternating minimization," in *IEEE ICIP*, Sept. 2013, pp. 69–73.

[18] J. Bezdek and R. Hathaway, "Some notes on alternating optimization," in *Advances in Soft Computing – AFSS 2002*, vol. 2275 of *Lecture Notes in Computer Science*, pp. 288–300. Springer, 2002.

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

[20] Laurent Condat, "Fast projection onto the simplex and the ℓ_1 ball," *Mathematical Programming*, vol. 158, no. 1, pp. 575–585, 2016.

[21] P. Rodriguez, "An accelerated Newton's method for projections onto the ℓ_1-ball," Submitted to *IEEE MLSP*, 2017.

[22] P. Rodríguez and B. Wohlberg, "incremental PCP simulations," `http://goo.gl/PlaEze`.

[23] Y. Wang, P. Jodoin, F. Porikli, J. Konrad, Y. Benezeth, and P. Ishwar, "Cdnet 2014: An expanded change detection benchmark dataset," in *IEEE CVPR*, June 2014, pp. 393–400.

[24] P. Rosin, "Unimodal thresholding," *Pattern Recognition*, vol. 34, pp. 2083–2096, 2001.