A MATLAB IMPLEMENTATION OF A FAST INCREMENTAL PRINCIPAL COMPONENT PURSUIT ALGORITHM FOR VIDEO BACKGROUND MODELING

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

In this proposal we present a Matlab-only implementation of a simple, novel, and fully incremental Principal Component Pursuit (PCP) algorithm for Video Background Modeling. Our implementation can process full HD color 1920×1088 videos at a rate of 0.61 seconds per frame running on a standard laptop (Intel i7-2670QM quad-core, 6GB RAM, 2.2 GHz).

Unlike other incremental or online PCP-like algorithms, such as ReProCS, GRASTA or pROST, the initialization stage of our implementation is extremely fast, has modest memory requirements (6.5 seconds and less than 0.5 Gb for a full HD video), and is also able to quickly adapt to changes in the background. Moreover our implementation can also process live-feed videos, which in our proposed demonstration will be acquired via a wireless camera, resulting in an interactive demonstration where the the moving objects to be segmented are the audience.

***Index Terms*—** Principal Component Pursuit, Video Background Modeling, incremental SVD

1. INTRODUCTION AND MOTIVATION

The Principal Component Pursuit (PCP) optimization problem is defined by

$$\arg\min_{L,S} \|L\|_* + \lambda\|S\|_1 \ \text{ s.t. } D = L + S \tag{1}$$

where $D \in \Re^{m \times n}$ 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), $L \in \Re^{m \times n}$ is a low rank (assume to be $r \ll m, n$) matrix representing the background, $S \in \Re^{m \times n}$ is a sparse matrix representing the foreground, $\|L\|_*$ is the nuclear norm of matrix L (*i.e.* $\sum_k |\sigma_k(L)|$), and $\|S\|_1$ is the ℓ^1 norm of S viewed as a column vector.

Although PCP provides state of the art performance [1, 2] in the video background modeling problem, it has several limitations [2], two prominent ones being its high computational cost and the fact that PCP is batch method: a large number of frames have to be observed before starting any processing, resulting in a large memory requirements, usually in the order of $1 \sim 10$ gigabytes.

There are some incremental or online PCP-like algorithms, such as ReProCS [3, 4, 5], GRASTA [6] or pROST [7] that address the above mentioned drawbacks. However, these approaches have their own limitations. ReProCS is not a real-time algorithm, and cannot process real videos where multiple moving objects enter and leave the field of view of the camera. GRASTA needs a temporally sub-sampled version of all the video frames during its initialization step, which can have a relative high complexity. The pROST algorithm has similar limitations.

For this ICIP'14 "Show & Tell" event we propose to demonstrate the Matlab implementation of a novel incremental PCP algorithm, which fundamentally differs from current state of the art PCP algorithms: (i) the proposed algorithm can process an input video in an incremental fashion; (ii) it can adapt to changes in the background; (iii) its computational cost is $O(14 \cdot m \cdot r)$ per frame; (iv) its memory footprint is $O(3 \cdot m \cdot r)$.

Moreover, since our implementation can processes a live-feed video (from a wireless camera), we plan to run such demonstration using a live-feed of the people passing by the facilities that will be used for the "Show & Tell" event.

2. SCIENTIFIC AND TECHNICAL DESCRIPTION

In this section we briefly summarize the mathematical methods used to develop the proposed incremental PCP algorithm.

2.1. Related work

In the derivation of our novel incremental PCP algorithm we have modified a recently published algorithm for (batch) PCP, called amFastPCP [8], to be able to handle incremental and

*There is currently a patent pending that covers the incremental PCP method described in this proposal.

†This research was supported by the NNSA's LaboratoryDirected Research and Development Program.

rank-1 modifications for thin SVD [9, 10, 11].

Instead of solving (1) directly, the amFastPCP algorithm solves the equivalent alternating minimization

$$L^{(j+1)} = \arg\min_{L} \quad \|L + S^{(j)} - D\|_F^2 \text{ s.t. } \mathrm{rank}(L) = r \quad (2)$$

$$S^{(j+1)} = \arg\min_{S} \quad \|L^{(j+1)} + S - D\|_F^2 + \lambda\|S\|_1, \quad (3)$$

where we note that sub-problem (2) can be solved by computing a partial (with r components) SVD of $D - S^{(j)}$. This particular structure of amFastPCP (with no auxiliary / dual variables involved, such as in the case of the inexact ALM algorithm [12] or other related algorithms) allows us to use incremental or rank-1 modifications to compute the solution of sub-problem (2) in an incremental fashion.

2.2. Technicalities of the Proposed Implementation

We assume that we have computed L_{k_0} (low-rank) and S_{k_0} (sparse), where $L_{k_0} + S_{k_0} = D_{k_0}$ and $D_{k_0} = D(:, 1 : k_0)$ and that we know the partial (thin) SVD of $L_{k_0} = U_r \Sigma_r V_r^T$, where $\Sigma_r \in \Re^{r \times r}$. Although this can be easily done by solving the batch PCP problem for D_{k_0} via any algorithm, here we mention that it is possible to used a slightly modified version of the procedure described below for this purpose.

If we were to solve the PCP problem when the next frame $\mathbf{d}_k$ ($k = k_0 + 1$) is available via the amFastPCP algorithm, then, due to the structure of (2), it is possible to find a solution via the incremental thin SVD procedure (non-increasing rank), assuming that the background does not change, or changes slowly. However, in a real scenario this condition will not hold for long; in this case we could use the downdate procedure (see [11]) to "forget" the background frames that are "too old" keeping a low-rank estimation of a (more or less) constant background, resulting in an incremental PCP algorithm that can adapt to changes in the background.

3. IMPLEMENTATION AND USE

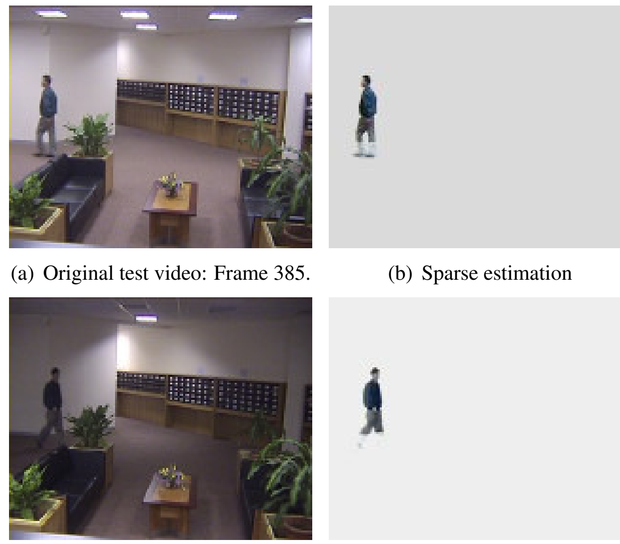
We have implemented our incremental PCP algorithm in Matlab (2013a), although we are currently working on a C implementation; the code of our implementation, which can process either pre-recorded videos or live-feed, is publicly available and can be downloaded from [13].

The demonstration we propose to present for this ICIP'14 "Show & Tell" event focus on two aspects: (i) processing of pre-recorded videos and (ii) processing of live-feeds. For the live-feed video demonstration we will use a wireless camera that acquires color (or grayscale) frames of 640×480 pixel, which can be processed at a rate of 8 frames per second (or 20 fps for grayscale, see Table 1). This live-feed processing will result in an interactive demonstration where the audience represent the moving objects to be segmented. Such a real-time demonstration at this frame size and frame rate would not be possible using any other existing PCP algorithms.

In Figs. 1 and 2 we show the sparse component estimated via our proposed method for for a 160×128 (dynamic background, also used in [6]) and 1920×1088 color video (from the Neovison2 dataset [14, Tower's 3rd video]) respectively. Moreover, other results can be observed in [13].

Test video	grayscale			color		
	Init.	Average	fps	Init.	Average	fps
160×128	0.22	0.0089	112.3	0.445	0.017	58
640×480	0.77	0.048	20.8	1.16	0.12	8
1920×1088	3.56	0.25	4	6.25	0.61	1.5

Table 1. Initialization and average processing time (seconds) per frame for our incremental PCP Matlab implementation.



(a) Original test video: Frame 385. (b) Sparse estimation

(c) Original test video: Frame 385. (d) Sparse estimation

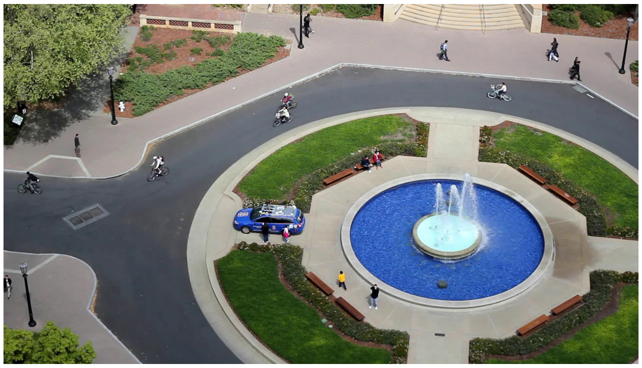
Fig. 1. Sparse video approximation for the 160×128 test (dynamic background) video via the proposed algorithm.

4. CONCLUSIONS AND FUTURE DEVELOPMENTS

The proposed method (for a complete description, see [15]) which has an extremely low memory footprint $O(3 \cdot m \cdot r)$ and a computational cost of $O(14 \cdot m \cdot r)$ per frame, can process full HD color videos at a rate of 0.61 seconds per frame, using a Matlab-only single-threaded implementation. Given the low memory footprint and complexity of the propose algorithm, the authors foresee that with an optimized implementation (e.g. parallel implementation based on CUDA technology), the proposed algorithm could improve its computational performance a hundred-fold; this would make it possible to analyze full HD videos in real-time.

5. REFERENCES

[1] C. Guyon, T. Bouwmans, and E. Zahzah, "Robust principal component analysis for background subtraction: Systematic evaluation and comparative analysis," in *Principal Component Analysis*. Ch. 12. InTech, 2012.



(a) Original test video: Frame 200.

(b) Original test video: Frame 400.

(c) Proposed algorithm sparse approximation: Frame 200.

(d) Proposed algorithm sparse approximation: Frame 400.

Fig. 2. Sparse video approximation for the 1980×1088 test video via the proposed algorithm.

[2] 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.

[3] C. Qiu and N. Vaswani, "Automated recursive projected cs (ReProCS) for real-time video layering," in *International Recognition (CVPR)*, 2012.

[4] C. Qiu and N. Vaswani, "Real-time robust principal components' pursuit," *CoRR*, vol. abs/1010.0608, 2010.

[5] C. Qiu and N. Vaswani, "Support predicted modified-cs for recursive robust principal components pursuit," in *IEEE Int. Symposium on Inf. Theory (ISIT)*, 2011.

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

[7] 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.

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

[9] Y. Chahlaoui, K. Gallivan, and P. Van Dooren, "An incremental method for computing dominant singular spaces," in *Computational Information Retrieval*, pp. 53–62. SIAM, 2001.

[10] C. Baker, K. Gallivan, and P. Van Dooren, "Low-rank incremental methods for computing dominant singular subspaces," *Linear Algebra and its Applications*, vol. 436, no. 8, pp. 2866 – 2888, 2012.

[11] 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.

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

[13] P. Rodriguez and B. Wohlberg, "Fast PCP simulation," `http://sites.google.com/a/istec.net/prodrig/Home/en/pubs/incpcp`.

[14] "USC Neovision2 Project," DARPA Neovision2, data available from `http://ilab.usc.edu/neo2/`.

[15] P. Rodríguez and B. Wohlberg, "Incremental principal component pursuit for video background modeling," submitted, Signal Processing Letters (SPL), 2014.