GHOSTING SUPPRESSION FOR INCREMENTAL PRINCIPAL COMPONENT PURSUIT ALGORITHMS

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 video background modeling, ghosting occurs when an object that belongs to the background is assigned to the foreground. In the context of Principal Component Pursuit, this usually occurs when a moving object occludes a high contrast background object, a moving object suddenly stops, or a stationary object suddenly starts moving.

Based on a previously developed incremental PCP method, we propose a novel algorithm that uses two simultaneous background estimates based on observations over the previous n_1 and n_2 ($n_1 \ll n_2$) frames in order to identify and diminish the ghosting effect. Our computational results show that the proposed method greatly improves both the subjective quality and accuracy as determined by the F-measure.

***Index Terms*—** Incremental Principal Component Pursuit, Ghosting

1. INTRODUCTION

Video background modeling consists of segmenting the moving objects or "foreground" from the static "background". Ghosting occurs when the foreground estimate includes phantoms or smear replicas from actual moving objects, or from objects that really belong to the background. An example of ghosting is shown in Fig. 1. Ghosting not only visually degrades the foreground estimation, but also has a negative impact when estimating a binary foreground mask, which is a key pre-processing task for target tracking, recognition and behavior analysis in digital videos.

The Principal Component Pursuit (PCP) method is currently considered to be one of the leading algorithms for video background modeling [1]. Although the PCP method is inherently a batch method (for a complete list of algorithms, see [1]) in that a large number of frames have to be observed before starting any processing, there exits a handful of incremental/on-line alternatives [2, 3, 4, 5, 6], for which the processing is performed one frame at a time. While these alternatives have some advantages over their batch counterparts, mainly their low computational cost and memory footprint, which could allow real-time processing of live-feed videos, they exhibit ghosting effects that are not usually observed in the PCP batch methods.

In this paper, we exploit a feature that is common in the incremental/on-line PCP-like algorithms[1]: the current background estimate is the result of observing a limited number of past frames. While the particular details of how this is handled by [2, 3, 4, 5, 6] varies, this feature allows these type of algorithms to adapt to changes in the background on the fly. The key and novel idea of the proposed ghosting suppression method is to simultaneously use two background estimates derived from the previous n_1 and n_2 frames respectively ($n_1 \ll n_2$) to reduce the ghosting effect.

Our computational results, which are carried out with the incPCP algorithm [6], show that our proposed algorithm efficaciously diminishes the ghosting effect observed in the foreground estimate and at the same time attains better accuracy (F-measure) than other incremental PCP algorithms when estimating a binary foreground mask, even when the mask is computed via a simple global thresholding per frame.

[1] Algorithms [2, 3, 4, 5, 6] use different terms to refer to the same general property; in this paper, from this point onwards, we choose to use the term "incremental".

2. PREVIOUS RELATED WORK

2.1. Ghost removal methods

Ghosting removal methods have attracted some attention over the past few years; in this section we give a brief overview of several works [7, 8, 9, 10, 11, 12, 13] that are based on different video background modeling approaches and that explicitly target the ghosting problem.

In the context of Gaussian Mixture Models (GMM), [10] proposed a method to detect ghosts and stationary foreground by dual-direction (observing "past" and "future" frames) background modeling. Instead, [8] incorporated an adaptive parameter adjustment to GMM. In both cases, superior results are obtained when compared to the traditional GMM.

Pixel-level methods, for which a model of the recent history is built for each pixel location, are also popular alternatives for video background modeling. In this context, [11] estimated the relevance of the pixel's historical background values, selectively adapting to background changes with different timescales and thus mitigating the ghosting effect. Instead, in [12] context features for each pixel were compressively sensed from local patches, and the background model was renewed in order to handle the ghosting effect. Adaptive median filtering and background update, based on the motion information, was used in [9] to remove the ghosting effect.

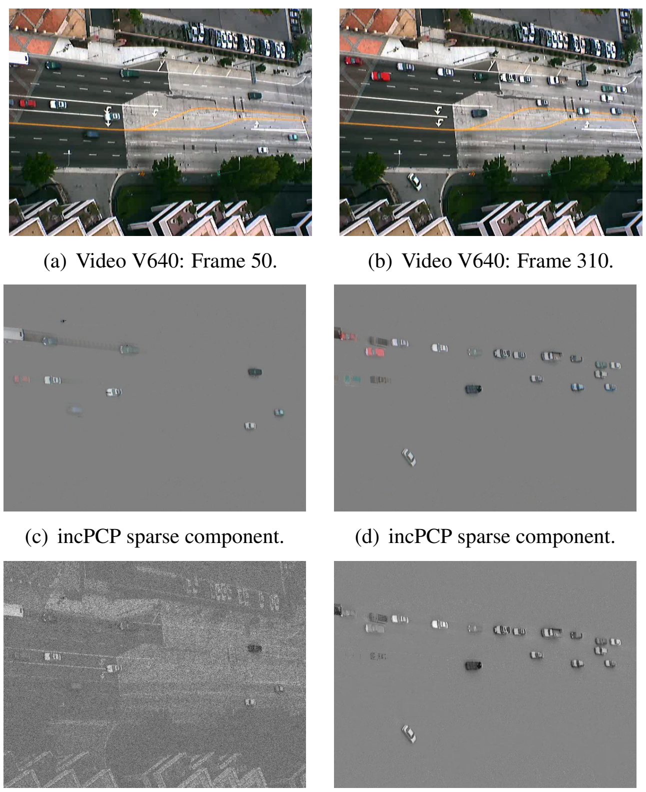
In the context of batch PCP-based methods, [13] proposed to minimize the partial sum of singular values, in place of the nuclear norm (see (1)); among other effects of this modification to the original problem, the authors claim that it delivers a ghost-free sparse representation.

In a more general context, [7] presented a fast and effective algorithm for ghost detection and removal, using edge comparison for ghost detection and removal during tracking. It is claimed that this method can be integrated into any video background modeling method that estimates a difference map between the current frame and a background estimates.

Finally we mention that there are several PCP-based methods (some of them incremental), see for instance [14, 15] among others, that usually include an extra foreground contiguity term which is used to directly estimate a binary foreground mask. However, such methods do not explicitly target the elimination of the ghost effect in the sparse component.

2.2. Incremental PCP methods

To the best of our knowledge, recursive projected compressive sensing (ReProCS) [2, 16] along with Grassmannian robust adaptive subspace tracking algorithm (GRASTA) [3], ℓ_p-norm robust online subspace tracking (pROST) [4], Grassmannian online subspace updates with structured sparsity (GOSUS) [5] and the incremental PCP (incPCP) [6] 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/recommended initial background estimate[2].



(a) Video V640: Frame 50. (b) Video V640: Frame 310.

(c) incPCP sparse component. (d) incPCP sparse component.

(e) GRASTA sparse component. (f) GRASTA sparse component.

Fig. 1. Original frames and corresponding sparse components for video V640[3] when analyzed with the incPCP[4] and GRASTA[5] algorithms. The ghosting effect is mainly noticeable in the upper left corner, where some vehicles are stopped for a while between frames 50 and 310.

2.3. Intuitive description of the incPCP algorithm

Since the proposed ghosting suppression method is implemented via the incPCP algorithm [20, 21, 6, 22, 23], here we give a succinct description of this algorithm. We first recall that the PCP problem

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

is derived as the convex relaxation of the original problem [24, Section 2]

$$\arg\min_{L,S} \quad \text{rank}(L) + \lambda\|S\|_0 \ \text{ s.t. } D = L + S \ , \tag{2}$$

based on decomposing matrix D such that $D = L + S$, with low-rank L (background) and sparse S (foreground). While most PCP algorithms, including the Augmented Lagrange Multiplier (ALM) and inexact ALM (iALM) algorithms [25, 26] are directly based on (1), this is not the only possible tractable problem that can be derived from (2). In particular, changing the constraint $D = L + S$ to a penalty, the rank penalty to an inequality constraint, relaxing the ℓ_0 norm by an ℓ_1 norm, leads to the problem

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

A computationally efficient solution can be found via an alternating optimization (AO) [27] procedure, since it seems natural to split (3) into a low-rank approximation, i.e. $\arg\min_L \frac{1}{2}\|L + S - D\|_F^2$ s.t. $\text{rank}(L) \leq r$, with fixed S, followed by a shrinkage, 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 the previously described AO procedure is of comparable quality to the solution of the original PCP problem (see [28] for details), being approximately an order of magnitude faster than the iALM [26] algorithm to construct a sparse component of similar quality.

Furthermore, the low-rank approximation sub-problem can be solved via a computationally efficient incremental procedure, based on rank-1 modifications for thin SVD ([29] and references therein), and thus (3) can also be easily solved incrementally, since the shrinkage step can be trivially computed in an incremental fashion.

In [6] it was shown that the resulting algorithm, called incPCP, is a fully incremental PCP algorithm for video background modeling, which is able to processes one frame at a time, obtaining similar results to batch PCP algorithms, while being able to adapt to changes in the background. Furthermore, in [6] extensive computational comparisons with state-of-the-art methods were also given.

3. PROPOSED ALGORITHM

In this section we first describe the general ideas for our propose ghost removal method, that could be adapted to any incremental PCP algorithm, and then proceed to give the particular adaptation of such ideas for the incPCP algorithm [6].

Given an incremental PCP algorithm, for any frame k, the low-rank ($\mathbf{l}$) and sparse ($\mathbf{s}$) components satisfy

$$\mathbf{d}_k \approx \mathbf{l}_k^{(n)} + \mathbf{s}_k^{(n)}, \tag{4}$$

where $\mathbf{d}_k$ is the current frame from the observed video; in (4) we use the super-script n to differentiate low-rank and sparse components

[2] 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 [6, Section 6]. pROST is closely related to GRASTA, and it shares the same restrictions. All variants of ReProCS also use a batch initialization stage.

[3] A 640×480 pixel, 400-frame color video sequence at 15 fps, from the Lankershim Boulevard traffic surveillance dataset [17, cam. 3]

[4] Matlab code is publicly available [18].

[5] Publicly available Matlab code [19] can only process grayscale videos.

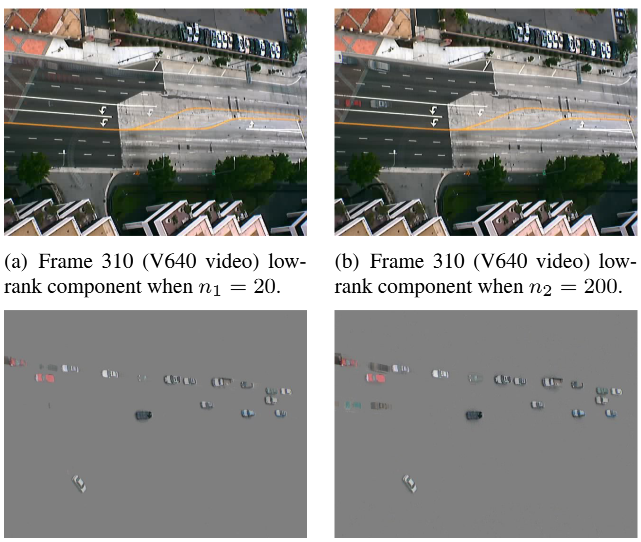
that have been obtained using the information of the past n frames. It is also worth recalling that in the context of PCP, once the low-rank component is available, the sparse component is computed via the soft-thresholding shrinkage

$$\mathbf{s}_k^{(n)} = \mathrm{shrink}(\mathbf{d}_k - \mathbf{l}_k^{(n)}, \lambda), \qquad (5)$$

where $\mathrm{shrink}(x, \lambda) = \mathrm{sign}(x) \max\{0, |x| - \lambda\}$.

Two simultaneous low-rank components, $\mathbf{l}_k^{(n_1)}$ and $\mathbf{l}_k^{(n_2)}$, with $n_1 \ll n_2$ will be different if a video event's interpretation differs over a longer/shorter time frame observation[6]: a typical example (albeit not the only one) would occur when an object that is considered "background" for a short time frame observation, is identified as a mobile object that swiftly stops or as a stationary object that swiftly starts moving when a larger time frame observation is considered.

Furthermore, the sparse components, $\mathbf{s}_k^{(n_1)}$ and $\mathbf{s}_k^{(n_2)}$, will reflect the above mentioned differences as ghosts, which would be more or less prominent depending on the level of intensity of the non-background objects that do appear on the low-rank component; this is depicted in Fig. 2.



(a) Frame 310 (V640 video) low-rank component when $n_1 = 20$.

(b) Frame 310 (V640 video) low-rank component when $n_2 = 200$.

(c) Frame 310 (V640 video) sparse component when $n_1 = 20$.

(d) Frame 310 (V640 video) sparse component when $n_2 = 200$.

Fig. 2. Two sets of low-rank / sparse components of frame 310, from the V640 test video, are shown. (a), (c) and (b), (d) are derived from the previous 20 and 200 frames respectively. Differences and ghosts are mainly observed in the upper left corner (see also Fig. 1).

If a foreground binary mask is estimated from each sparse component, i.e. $\mathbf{m}_k^{(n_1)}$ and $\mathbf{m}_k^{(n_2)}$, these masks will include the moving objects as well as ghosts. However, as can be surmised from Fig. 2, the intersection of these binary masks will, with high confidence, only include the moving objects; likewise, $\mathcal{B}_k = \sim \mathbf{m}_k^{(n_1)} \cup \sim \mathbf{m}_k^{(n_2)}$, the union of the masks complement, will include all the pixels of the background that are not occluded by a moving object.

Provided that the previous statement is true, then $\mathcal{B}_k$ can be used (i) to generate a spatially varying λ value for frame k (oppose to a fixed global value as used in (5)) to heavily penalize the high confidence background pixels, (ii) to generate a "new" input frame $\hat{\mathbf{d}}_k^{(n)} = \mathbf{d}_k \odot \mathcal{B}_k + \mathbf{l}_k^{(n)} \odot (\mathbf{1} - \mathcal{B}_k)$, where $\odot$ is element-wise multiplication (Hadamard product), in order to replace the effect of the previously processed original frame $\mathbf{d}_k$. It is worth noting that the previously described actions, implicitly assume that a particular incremental PCP algorithm has the ability to "forget" or "replace" the effect of a given frame in the low-rank component, or has the ability to feedback an improved background estimate.

As mentioned in section 2.3, the incPCP algorithm [6], makes use of rank-1 modifications for thin SVD, which includes the "update", "replace" and "downdate" ("forget") operations (see [29] for details). Based on such operations, in Algorithm 1 we describe the specific details on how to implement the above mentioned ideas for the incPCP algorithm, where $[U_k^{(n)}, \Sigma_k^{(n)}, V_k^{(n)}]$ represents the SVD decomposition of the low-rank component as observed for the last n frames, $\mathrm{dwnSVD}(.)$ and $\mathrm{incSVD}(.)$ represent the "downdate" and "update" operations.

Inputs : observed video frame $\mathbf{d}_k$, regularization parameter λ, low-rank components $\mathbf{l}_k^{(n_1)}$ and $\mathbf{l}_k^{(n_2)}$, sparse components $\mathbf{s}_k^{(n_1)}$ and $\mathbf{s}_k^{(n_2)}$, scalar $\alpha > 1$.

Compute: considering $n = \{n_1, n_2\}$

$\mathbf{m}_k^{(n)} = \mathrm{mask}(\mathbf{s}_k^{(n)})$ (see comments in Section 4)

$\mathcal{B}_k = \sim \mathbf{m}_k^{(n_1)} \cup \sim \mathbf{m}_k^{(n_2)}$

$\hat{\mathbf{d}}_k^{(n)} = \mathbf{d}_k \odot \mathcal{B}_k + \mathbf{l}_k^{(n)} \odot (\mathbf{1} - \mathcal{B}_k)$

$\boldsymbol{\lambda}_k = \lambda \cdot (\mathbf{1} - \mathcal{B}_k) + \alpha \cdot \lambda \cdot \mathcal{B}_k$ (spatially varying λ value)

$[U_k^{(n)}, \Sigma_k^{(n)}, V_k^{(n)}] = \mathrm{dwnSVD}(\text{"last col."}, U_k^{(n)}, \Sigma_k^{(n)}, V_k^{(n)})$

$[U_k^{(n)}, \Sigma_k^{(n)}, V_k^{(n)}] = \mathrm{incSVD}(\hat{\mathbf{d}}_k^{(n)}, U_k^{(n)}, \Sigma_k^{(n)}, V_k^{(n)})$

$\mathbf{l}_k^{(n)} = U_k^{(n)} \cdot \Sigma_k^{(n)} \cdot V_k^{(n)}(end, :)^T$

$\mathbf{s}_k^{(n)} = \mathrm{shrink}(\mathbf{d}_k - \mathbf{l}_k^{(n)}, \boldsymbol{\lambda}_k)$

Algorithm 1: Ghosting suppression incremental PCP (gs-incPCP).

4. COMPUTATIONAL RESULTS

We present F-measure based accuracy results (see Table 1) for the I2R dataset[7] [30] and from two challenging videos[8] from the CDnet dataset [31]. The F-measure, which makes use of a binary ground-truth, is defined as

$$F = \frac{2 \cdot P \cdot R}{P + R}, \quad P = \frac{TP}{TP + FN}, \quad R = \frac{TP}{TP + FP} \qquad (6)$$

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.

In order to compute the F-measure for incPCP (original and ghosting suppression variants) as well as for GRASTA and GOSUS, a threshold is needed to compute the binary foreground mask. For the results presented in this section, this threshold has been computed via an automatic unimodal segmentation [32] since the absolute value of the sparse representation has an unimodal histogram. This approach, although simple, adapts its threshold for each sparse representation and ensures that all algorithms are fairly treated.

[6] Here we are ruling out dramatic changes in the background, such when the camera is moved, when a sudden illumination change occurs, etc.

[7] The number of available ground-truth frames is 20 for all cases.

[8] 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.

Video	F-measure					
	grayscale			color		
	incPCP	gs-incPCP	GRASTA	incPCP	gs-incPCP	GOSUS
Bootstrap – I2R ($120 \times 160 \times 3057$, crowd scene)	0.587	**0.611**	0.608	0.636	**0.669**	0.659
Campus – I2R ($120 \times 160 \times 1439$, waving trees)	0.244	**0.771**	0.215	0.281	**0.821**	0.166
Curtain – I2R ($120 \times 160 \times 2964$, waving curtain)	0.741	0.757 (n_1=100)	**0.787**	0.758	0.783 (n_1=100)	**0.870**
Escalator – I2R ($130 \times 160 \times 3417$, moving escalator)	0.481	**0.627**	0.539	0.472	**0.622**	0.405
Fountain – I2R ($128 \times 160 \times 523$, fountain water)	0.627	**0.769**	0.662	0.632	**0.789**	0.677
Hall – I2R ($144 \times 176 \times 3548$, crowd scene)	0.570	0.601	**0.625**	0.609	**0.677**	0.464
Lobby – I2R ($128 \times 160 \times 1546$, switching light)	0.550	0.466 (n_1=150)	**0.567**	**0.713**	0.657 (n_1=150)	0.185
Mall – I2R ($256 \times 320 \times 1286$, crowd scene)	0.693	**0.718**	0.692	0.746	**0.772**	0.715
WaterSurface – I2R ($128 \times 160 \times 633$, water surface)	0.636	**0.818**	0.772	0.632	**0.829**	0.787
V320 – CDnet ($320 \times 240 \times 1700$, highway camera)	0.745	**0.864**	0.773	0.794	**0.891**	0.549
V720 – CDnet ($720 \times 576 \times 1200$, train station)	0.687	**0.765**	0.169	0.728	**0.802**	0.426

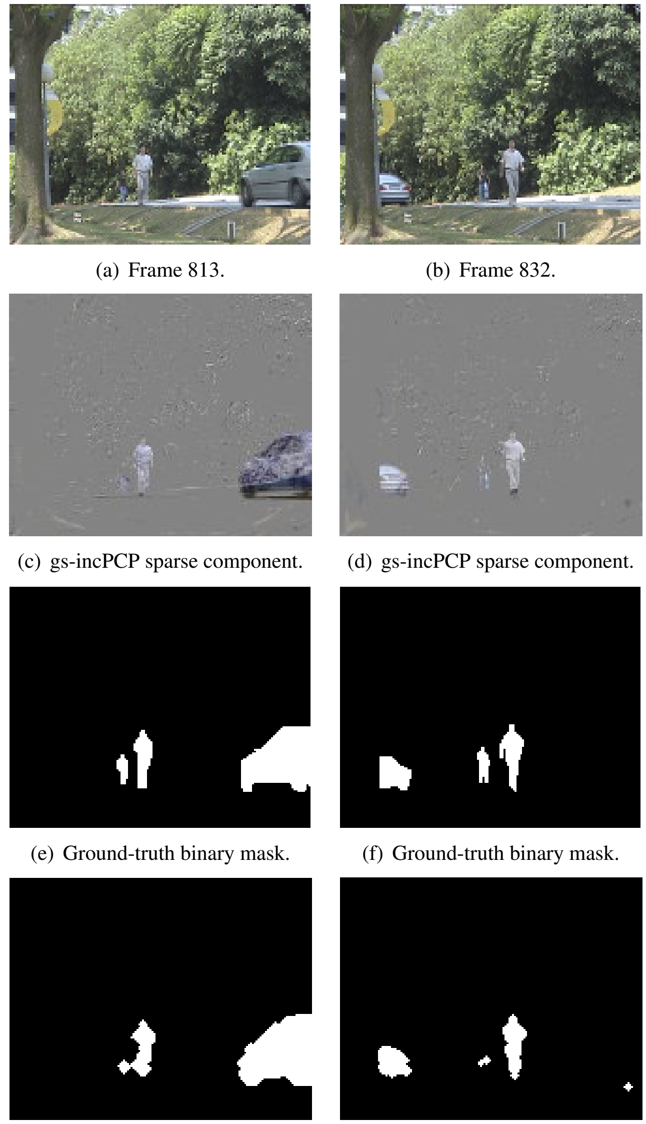
Table 1. Accuracy performance via the F-measure on the I2R and CDnet datasets for the incPCP, original and ghosting suppression (gs-incPCP, Matlab code available [18]) variants, GRASTA and GOSUS algorithms. Below each video's name we include the size and total number of frames (rows $\times$ columns $\times$ frames) along with a short description. The bold values are the largest F-measure values (grayscale and color cases are treated independently).

Unless otherwise noted, the ghosting suppression incPCP algorithm uses[9] $n_1 = 20$, $n_2 = 200$ and $\alpha = 2$ (see Algorithm 1). Furthermore, for the GRASTA and GOSUS algorithms we use their default batch procedure for the initial background estimation, since they gives the best F-measure results.

The accuracy results for the I2R and CDnet datasets, listed in Table 1, show that the proposed method (gs-incPCP, Matlab code available [18]) gives superior performance when compared to the considered alternatives. Moreover, gs-incPCP is particularly effective when the analyzed video has an moving background, such as in the case of (i) the "Campus"[10] and "V320" videos, where waving trees/leaves are observed, (ii) the "WaterSurface" video, where the background is mainly waving ocean, (iii) the "Fountain" video, where a large fountain waterfall is located behind a walkway, and (iv) the "Escalator" video, where a large part of the scene is an escalator used by people passing by.

[9]As a rule of thumb $n_2 = 10 \times n_1$. The value of n_1 depends on video event's interpretation over a short time frame observation; for most of the considered test videos "short time" is about one second, and thus $n_1 = 20$.

[10]For the "Campus" test video, the gs-incPCP's F-measure is largely better than those of all alternatives. In Fig. 3 we depict some of the related results.



(a) Frame 813. (b) Frame 832.

(c) gs-incPCP sparse component. (d) gs-incPCP sparse component.

(e) Ground-truth binary mask. (f) Ground-truth binary mask.

(g) Estimated binary mask. (h) Estimated binary mask.

Fig. 3. Original 813 (a) and 832 (b) frames of the "Campus" video, along with the corresponding ground-truth binary masks (e)-(f) and estimated masks (g)-(h) and sparse component (c)-(d) via the gs-incPCP algorithm.

5. CONCLUSIONS

The proposed method, ghosting suppression incremental PCP, is an effective algorithm that identifies and diminishes ghosting artifacts, including those in videos with a moving background, e.g. (i) waving trees and ocean, such as in the "Campus", "WaterSurface" and "V320" test videos, (ii) for repetitive movement, such as in the "Fountain" and "Escalator" test videos. The proposed method gives superior quality and accuracy as determined by the F-measure when estimating a binary foreground mask which is computed via a simple global thresholding per frame.

The proposed method is approximately four to five times slower than baseline incPCP, however it can attain a processing frame rate throughput of $3 \sim 10$ f.p.s. for the considered test videos on an Intel i7-4710HQ (2.5 GHz, 6MB Cache, 32GB RAM) based laptop.

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] C. Qiu and N. Vaswani, "Support predicted modified-CS for recursive robust principal components pursuit," in *IEEE Int'l Symposium on Information Theory*, 2011.

[3] 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.

[4] 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.

[5] J. Xu, V. Ithapu, L. Mukherjee, J. Rehg, and V. Singh, "GOSUS: Grassmannian online subspace updates with structured-sparsity," in *IEEE Int'l, Conf. on Comp. Vis.*, Dec. 2013, pp. 3376–3383.

[6] P. Rodriguez and B. Wohlberg, "Incremental principal component pursuit for video background modeling," *J. of Mathematical Imaging and Vision*, vol. 55, no. 1, pp. 1–18, 2016.

[7] F. Yin, D. Makris, and S. Velastin, "Time efficient ghost removal for motion detection in visual surveillance systems," *Electronics Letters*, vol. 44, no. 23, pp. 1351–1353, November 2008.

[8] P. Suo and Y. Wang, "An improved adaptive background modeling algorithm based on gaussian mixture model," in *Int'l Conf. on Signal Proc.*, Oct 2008, pp. 1436–1439.

[9] H. Yang, Y. Nam, W. Cho, and Y. Choi, "Adaptive background modeling for effective ghost removal and robust left object detection," in *Int'l Conf. on Information Technology Convergence and Services*, Aug 2010, pp. 1–6.

[10] G. Chuan, W. Yanjiang, and Q. Yujuan, "Ghosts and stationary foreground detection by dual-direction background modeling," in *IEEE Int'l Conf. on Signal Proc.*, Oct 2012, vol. 2, pp. 1115–1118.

[11] B. Wang and P. Dudek, "A fast self-tuning background subtraction algorithm," in *IEEE Conf. on Comp. Vis. and Pattern Recognition*, June 2014, pp. 401–404.

[12] L. Yang, H. Cheng, J. Su, and X. Li, "Pixel-to-model distance for robust background reconstruction," *IEEE Trans, on Circuits and Sys. for Video Technology*, 2015.

[13] T. Oh, Y. Tai, J. Bazin, H. Kim, and I. Kweon, "Partial sum minimization of singular values in robust pca: Algorithm and applications," *IEEE TPAMI*, 2015.

[14] X. Zhou, C. Yang, and W. Yu, "Moving object detection by detecting contiguous outliers in the low-rank representation," *IEEE TPAMI*, vol. 35, pp. 597–610, 2013.

[15] Y. Hu, K. Sirlantzis, G. Howells, N. Ragot, and P. Rodriguez, "An online background subtraction algorithm using a contiguously weighted linear regression model," in *European Signal Proc. Conf.*, Aug. 2015, pp. 1845–1849.

[16] 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.

[17] "Lankershim boulevard dataset," Jan. 2007, U.S. Department of Transportation Publication FHWA-HRT-07-029, data available from `http://ngsim-community.org/`.

[18] P. Rodríguez and B. Wohlberg, "incremental PCP simulations," `https://goo.gl/QNCxaf`.

[19] J. He, L. Balzano, and A. Szlam, "GRASTA code," `https://goo.gl/G41UC6`.

[20] P. Rodriguez and B. Wohlberg, "A Matlab implementation of a fast incremental principal component pursuit algorithm for video background modeling," in *IEEE Int'l. Conf. on Image Proc.*, Oct. 2014, pp. 3414–3416.

[21] P. Rodriguez and B. Wohlberg, "Video background modeling under impulse noise," in *IEEE Int'l. Conf. on Image Proc. (ICIP)*, Paris, France, Oct. 2014, pp. 1041–1045.

[22] P. Rodriguez and B. Wohlberg, "Translational and rotational jitter invariant incremental principal component pursuit for video background modeling," in *IEEE Int'l Conf. on Image Proc.*, Sept 2015, pp. 537–541.

[23] G. Silva and P. Rodriguez, "Jitter invariant incremental principal component pursuit for video background modeling on the TK1," in *Asilomar Conf. on Signals, Systems, and Computers*, Nov 2015, pp. 1403–1407.

[24] 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.*, 2009, pp. 2080–2088.

[25] G. Liu, Z. Lin, and Y. Yu, "Robust subspace segmentation by low-rank representation.," in *Int'l. Conf. Mach. Learn. (ICML)*, 2010, pp. 663–670.

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

[27] 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 Berlin Heidelberg, 2002.

[28] P. Rodriguez and B. Wohlberg, "Fast principal component pursuit via alternating minimization," in *IEEE Int'l. Conf. on Image Proc.*, Sept. 2013, pp. 69–73.

[29] 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.

[30] L. Li, W. Huang, I. Gu, and Q. Tian, "Background model test data," `http://goo.gl/pC3ge8`.

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

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