2016 IEEE Statistical Signal Processing Workshop (SSP)

# VIDEO DENOISING VIA ONLINE SPARSE AND LOW-RANK MATRIX DECOMPOSITION

*Han Guo and Namrata Vaswani*

ECE dept, Iowa State University, Ames, IA, USA
Email: {hanguo,namrata}@iastate.edu

## ABSTRACT

Video denoising refers to the problem of removing "noise" from a video sequence. Here the term "noise" is used in a broad sense to refer to any corruption or outlier or interference that is not the quantity of interest. In this work, we develop a novel approach to video denoising that is based on the idea that most noisy or corrupted videos can be split into two parts - the approximate "low-rank" layer and the "sparse layer". We first splitting the given video into these two layers, and then apply an existing state-of-the-art denoising algorithm on each layer. We show, using extensive experiments, that our denoising approach outperforms the state-of-the art denoising algorithms.

***Index Terms—*** denoising, sparse and low-rank matrix

## 1. INTRODUCTION

Video denoising refers to the problem of removing "noise" from a video sequence. Here the term "noise" is used in a broad sense to refer to any corruption or outlier or interference that is not the quantity of interest.

In the last few decades there has been a lot of work on video denoising. Many of the approaches extend image denoising ideas to 3D by also exploiting dependencies across the temporal dimension. An important example of this is "grouping and collaboratively filtering" approaches which try to search for similar image patches both within an image frame and across nearby frames, followed by collaboratively filtering the noise from the stack of matched patches [1, 2, 3, 4, 5]. One of the most effective methods for image denoising, Block Matching and 3D filtering (BM3D) [2], is from this category of techniques. In BM3D, similar image blocks are stacked in a 3D array followed by applying a noise shrinkage operator in a transform domain. In its video version, VBM3D [6], the method is generalized to video denoising by searching for similar blocks across multiple frames. Another related work [7, 8] either uses matrix completion [7] or sparse and low rank matrix approximation [8] on grouped image patches to remove outliers.

Other recent works on video denoising include approaches that use motion compensation algorithms from the video compression literature followed by denoising of similar nearby blocks [9, 10]; and approaches that use wavelet transform based [11, 12, 13] and discrete cosine transform (DCT) based [14] denoising solutions. Very recent video denoising methods include algorithms based on learning a sparsifying transform [15, 16, 17, 18].

Within the image denoising literature, the most promising recent approaches are based on deep learning [19, 20, 21].

In this paper, we develop a novel approach to video denoising, called *Layering Denoising*, that is based on the idea that most noisy or corrupted videos can be split into two parts - the approximate "low-rank" layer and the "sparse layer". Here "low-rank" layer means that the matrix formed by each image of this layer arranged as a column vector is low-rank. The "sparse" layer is similarly defined. The low-rank plus sparse assumption for corrupted videos can be seen in many denoising scenarios:

- In traditional denoising scenario, consider videos that are corrupted either by very large variance Gaussian noise, or a combination of salt-and pepper noise (or other impulsive noise) and small variance Gaussian noise. For these type of videos, the large magnitude part of the noise, together with the foreground person / object in the video, form the "sparse layer", while the background (slowly-changing in many applications, e.g., video surveillance) with small noise approximately forms the "low-rank" layer. The goal is to remove the Gaussian or mixed noise.

- In low-light videos of moving targets/objects, the denoising goal is to "see" the barely visible moving targets (sparse), from the slowly-changing dark background, which is low-rank and is considered as noise.

- In background editing applications, the slowly-changing background is corrupted by sparse foreground objects (outliers), and the denoising goal is to only recover the background.

Our algorithm proceeds by first splitting the given video into "sparse layer" and "low-rank layer". This is followed by applying an existing state-of-the-art denoising algorithm on each layer. For separating the video, we use a combination of an existing batch technique called principal components' pursuit (PCP) [22] and a recently proposed online and dynamic technique called Recursive Projected Compressive Sensing (ReProCS) [23, 24]. We initialize using PCP and use ReProCS afterwards (since, as shown in earlier work [24], it is faster and has better separation performance for videos where the sparse layer is either correlated over time or has large support size). This is followed by VBM3D on each of the two layers. In doing this, VBM3D exploits the specific characteristics of each layer and is able to find more matched blocks to filter over.

The performance of our algorithm is compared both with some of the best solutions from existing sparse + low-rank decomposition literature [25, 26, 27, 28, 29, 30, 31] as well as with just using VBM3D directly on the video. We also compare with one recent deep learning based image denoising solution [19] and an approach that makes use of sparse and low-rank matrix approximation on grouped image patches [8]. We provide comparisons for two denoising scenarios. The first is traditional denoising scenario, i.e., to remove add-on noise from the video. The noise can be Gaussian or a combination of Gaussian and salt and pepper noise. The second denoising scenario is to "see" object in the low-light environment. We show that our algorithm can label the moving object as the sparse layer.

### 1.1. Notation

In this paper, $P$ = approx-basis$(\mathcal{M}, b\%)$ means that $P$ is the $b\%$ left singular vectors' matrix for $\mathcal{M}$. PCP$(\mathcal{M})$ implements the PCP algorithm on data $\mathcal{M}$. Std-est$(\mathcal{M})$ denotes estimating the standard deviation from $\mathcal{M}$. VBM3D$(\mathcal{M}, \sigma)$ implements the VBM3D algorithm on data $\mathcal{M}$ with input standard deviation $\sigma$.

### 1.2. Problem Formulation

We denote the observed video frames at time $t$ by $M_t$. In general, $M_t = \text{Img}_t + W_t$, where $\text{Img}_t$ denotes the clean image and $W_t$ is the add-on noise, e.g., i.i.d. Gaussian or mixed noise. We write $M_t = S_t + L_t + W_t$ where $S_t$ is sparse, modeling the foreground object and $L_t$ lies in a low-dimensional subspace modeling the slowly-changing background. The denoising goal can be recovering $\text{Img}_t$, recovering $S_t$ or recovering $L_t$.

## 2. THE LAYERING-DENOISING ALGORITHM

We explain the algorithm as following:

**Initialization.** Take a batch of initial video frames $[M_1, M_2, \cdots, M_{t_\text{train}}]$ as training data and use stable-PCP [32] to separate it into a sparse matrix and a low-rank matrix $[L_1, L_2, \cdots, L_{t_\text{train}}]$. Do SVD on $[L_1, L_2, \cdots, L_{t_\text{train}}]$ and retain the top left singular vectors, denoted by $\hat{P}$.

**Splitting phase.** For $t \geq t_\text{train} + 1$, we split $M_t$ into two layers $\hat{S}_t$ and $\hat{L}_t$. Firstly, $M_t$ is projected to the space orthogonal to range$(\hat{P})$ to get the projected measurement vector,

$$y_t := (I - \hat{P}\hat{P}')M_t = \Phi M_t. \tag{1}$$

If we rewrite $W_t$ as $W_t = O_t + N_t$ where entries of $O_t$ have large magnitude comparable to $S_t$, and $N_t$ is the remaining, $y_t$ can be expressed as

$$y_t = \Phi(S_t + O_t) + \beta_t \text{ where } \beta_t := \Phi(L_t + N_t). \tag{2}$$

We solve

$$\min_x \|x\|_1 \text{ s.t. } \|y_t - \Phi x\|_2 \leq \xi \tag{3}$$

and denote its solution by $\hat{S}_t$ which is an approximation of $S_t + O_t$. $\hat{L}_t$ is obtained by simply subtracting $\hat{S}_t$ from $M_t$.

**Denoising phase.** We perform VBM3D on $\hat{S}_t$ and $\hat{L}_t$. Note that VBM3D here can be replaced with any other denoisers based on need. Their input standard deviations are estimated from $[\hat{S}_t, \ldots, \hat{S}_{t_\text{train}}]$ and $[\hat{L}_t, \ldots, \hat{L}_{t_\text{train}}]$ respectively. The final denoised output is simply the sum of the denoised layer $\hat{S}_t$ and $\hat{L}_t$, i.e., $I_{\text{denoised},t} = S_{\text{denoised},t} + L_{\text{denoised},t}$, or just one of $I_{\text{denoised},t}$ and $L_{\text{denoised},t}$ depends on the application.

**Subspace Update phase (Optional).** In long videos the span of the $L_t$'s will change with time. Hence one needs to update the subspace estimate $\hat{P}$ every so often. This can be done efficiently using the projection-PCA algorithm proposed in [24].

## 3. EXPERIMENTS

### 3.1. Removing Add-on Noise

In this part we compare our proposed denoise scheme with VBM3D, a deep learning image denoising method called Multi Layer Perceptron (MLP) [19], and an algorithm that uses sparse and low-rank matrix approximation on grouped patches [8] which we call SLMA for short, on several video datasets that can be downloaded from `http://www.ece.iastate.edu/~hanguo/denoise.html`. The denoised output of Algorithm 1 here refers to $\mathcal{I}_\text{denoised} = \mathcal{S}_\text{denoised} + \mathcal{L}_\text{denoised}$. The codes for algorithms being compared are downloaded from the authors' webpage. The available MLP code contains parameters that are trained solely from image patches that were corrupted with Gaussian noise with $\sigma = 25$ and hence the denoising performance is best with $\sigma = 25$ and deteriorates for other noise levels.

We present a thorough comparison on the Curtain and Lobby dataset, and summarize comparisons on other datasets in Table 1. The noise being added to the original image frames are Gaussian ($\sigma = 25$), Gaussian ($\sigma = 25$) plus salt and pepper noisy, and Gaussian ($\sigma = 70$). The input $\sigma$ for all algorithms is estimated from the noisy data rather then given the

**Algorithm 1** Layering-Denoising

**Input:** $M_1, M_2, \ldots, M_{t_{\text{max}}}$;

**Initialization:**

Obtain $[L_1, \ldots, L_{t_{\text{train}}}]$ from PCP($[M_1, \ldots, M_{t_{\text{train}}}]$);

$\hat{P} \leftarrow$ approx-basis$(\frac{1}{\sqrt{t_{\text{train}}}}[L_1, \ldots, L_{t_{\text{train}}}], 95\%)$

**Phase 1: Split $M_t$ into two layers**

1. $\hat{\sigma} \leftarrow$ Std-est$([M_1, \ldots, M_{t_{\text{train}}}])$
2. Apply ReProCS on $M_t$:
   (a) Compute $\hat{S}_t$ as the solution of simple-$\ell_1$ minimization (3).
   (b) $\hat{L}_t \leftarrow M_t - \hat{S}_t$

**Phase 2: Denoising**

1. $\hat{\sigma}_{\text{fg}} \leftarrow$ Std-est$([\hat{S}_t, \ldots, \hat{S}_{t_{\text{train}}}])$
   $\hat{\sigma}_{\text{bg}} \leftarrow$ Std-est$([\hat{L}_t, \ldots, \hat{L}_{t_{\text{train}}}])$
2. $\mathcal{S}_{\text{denoised}} \leftarrow$ VBM3D$([\hat{S}_1, \ldots, \hat{S}_{t_{\text{max}}}], \hat{\sigma}_{\text{fg}})$
   $\mathcal{L}_{\text{denoised}} \leftarrow$ VBM3D$([\hat{L}_1, \ldots, \hat{L}_{t_{\text{max}}}], \hat{\sigma}_{\text{bg}})$
3. $\mathcal{I}_{\text{denoised}} = \mathcal{S}_{\text{denoised}} + \mathcal{L}_{\text{denoised}}$

**Output:** $\mathcal{I}_{\text{denoised}}, \mathcal{S}_{\text{denoised}}, \mathcal{L}_{\text{denoised}}$

true value. We computed the frame-wise PSNRs for each case in Fig. 4 and Fig. 5 and show sample visual comparisons in Fig. 1 and Fig. 2. It can be seen in Fig. 4 and Fig. 5 that Algorithm 1 outperforms all other algorithms in all three noise level – the PSNR is the highest in almost all image frames. Visually, Algorithm 1 is able to recover more details of the images while other algorithms either fail or cause severe blurring effect.

We present the comparison on other datasets in Table 1. The noise being added to the original video is Gaussian, with standard deviation $\sigma$ increases from 25 to 70. As can be seen in the table, Algorithm 1 outperforms other algorithms as well.

### 3.2. Denoising in Low-light Environment

Here we test the performance of denoiseing in low-light environment, i.e., to see target signal in the low-light environment. The video was taken in a dark environment where a barly visible person walked throught the hallway. The algorithms being compared are foreground/background separation techniques – PCP, RSL [25], GRASTA [26], background subtraction, and video denoising approach VBM3D. The output we are using here is $S_{\text{denoised},t}$ and only its support (properly thresholded) is shown for ease of display. Fig. 3 shows that VBM3D is not capable of observing the walking person while our algorithm 1 can successfully mark the interested foreground person and the result is slightly better than or as good as other foreground/background separation techniques.

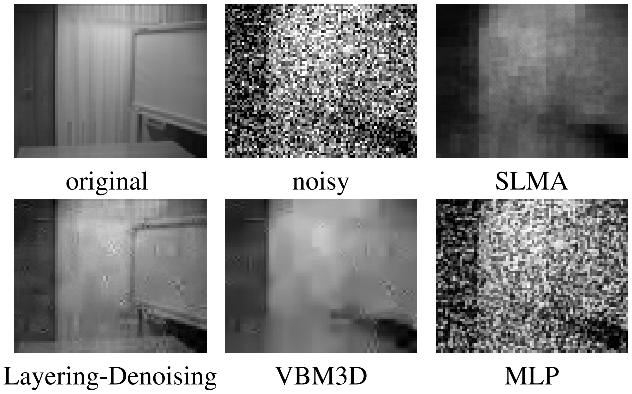


**Fig. 1**. Visual comparison of denoising performance for the Curtain dataset for very large Gaussian noise ($\sigma = 70$)

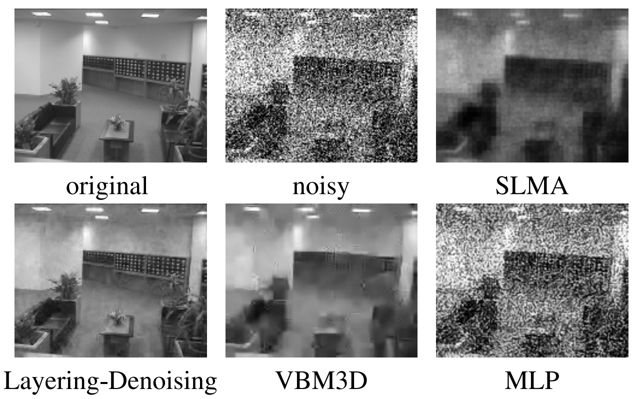


**Fig. 2**. Visual comparison of denoising performance for the Lobby dataset for very large Gaussian noise ($\sigma = 70$)

## 4. CONCLUSION

In this paper we develop a denoising scheme which enhances the denoising performance of the state-of-the-art algorithm VBM3D, and is able to achieve denoising in a broad sense – observing interested signal in low-light environment or filtering out large but sparse noise from slow changing videos followed by using of VBM3D to remove the small noise. In general, splitting the video first results in a relatively clean low-rank layer since the large noise goes to the sparse layer in the $\ell_1$ minimization (3). The clean layer improves the "grouping" accuracy in VBM3D. In fact one can replace VBM3D with other denoisers (procedures) to deal with specific noise models.

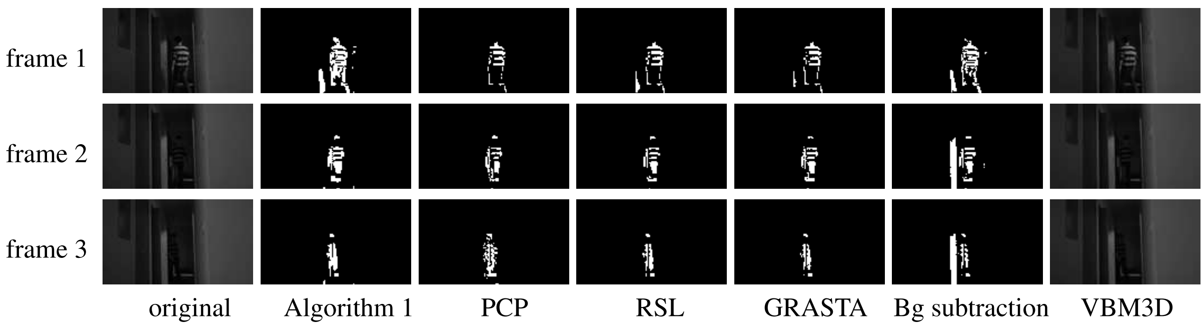


**Fig. 3**. Ability of "seeing" in the dark for three sample frames. From left to right: orignal dark image, results by Layering Denoising, PCP, RSL GRASTA, background subtraction and VBM3D.

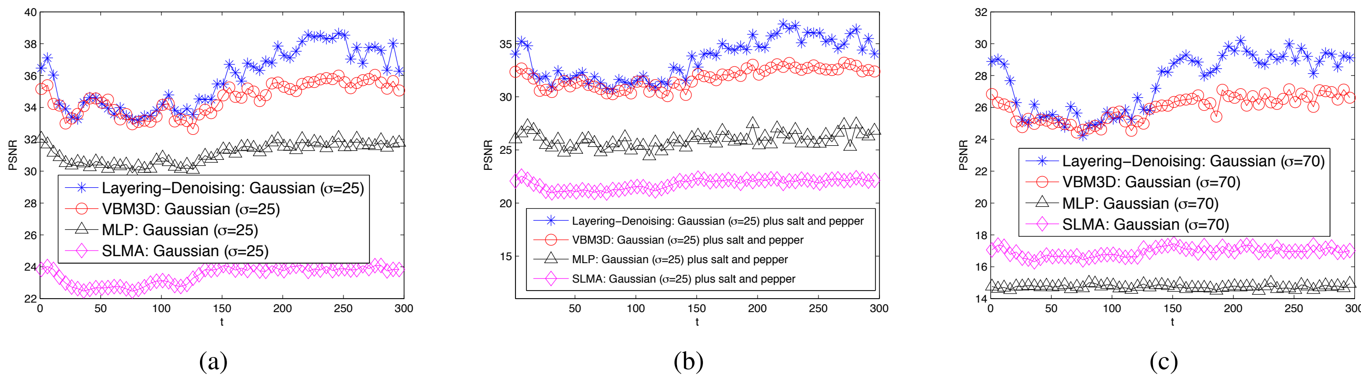


**Fig. 4**. Frame-wise PSNR for Curtain dataset with different noise level: (a) Gaussian noise with $\sigma = 25$, (b) Gaussian noise ($\sigma = 25$) plus salt and pepper noise, (c) Gaussian noise with $\sigma = 70$.

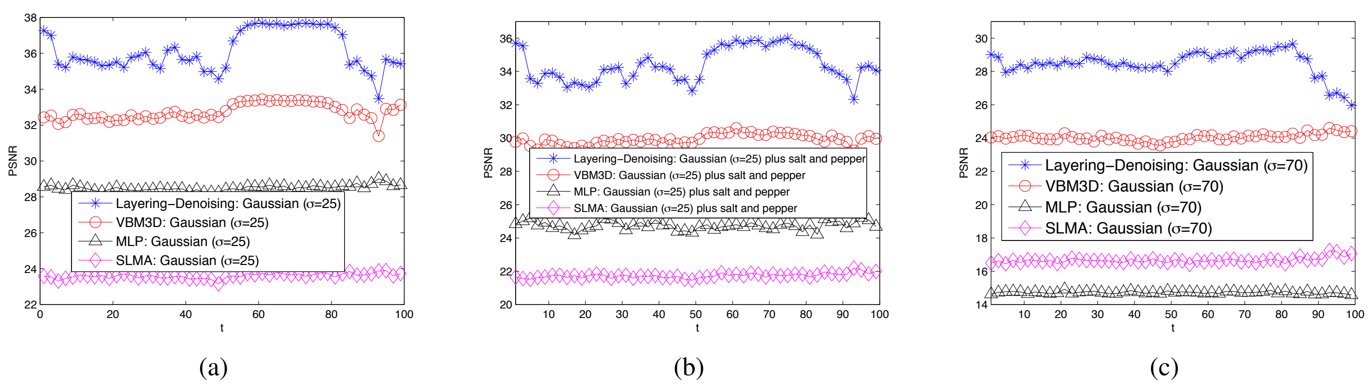


**Fig. 5**. Frame-wise PSNR for Lobby dataset with different noise level: (a) Gaussian noise with $\sigma = 25$, (b) Gaussian noise ($\sigma = 25$) plus salt and pepper noise, (c) Gaussian noise with $\sigma = 70$.

| $\sigma$ | fountain | | | | airport | | | |
|---|---|---|---|---|---|---|---|---|
| | Layering-Denoising | VBM3D | MLP | SLMA | Layering-Denoising | VBM3D | MLP | SLMA |
| 25 | 31.22 | 31.11 | 26.86 | 22.93 | 30.28 | 30.12 | 25.53 | 21.17 |
| 30 | 30.62 | 30.08 | 25.67 | 21.85 | 29.63 | 29.08 | 24.54 | 20.49 |
| 50 | 28.61 | 25.99 | 18.53 | 18.55 | 27.59 | 24.64 | 18.83 | 17.98 |
| 70 | 26.19 | 22.08 | 14.85 | 16.25 | 24.87 | 20.25 | 15.20 | 15.90 |

**Table 1**. Comparison of denoising performance: PSNR

## 5. REFERENCES


[1] Antoni Buades, Bartomeu Coll, and Jean-Michel Morel, "Image denoising by non-local averaging," in *Acoustics, Speech, and Signal Processing, 2005. Proceedings.(ICASSP'05). IEEE International Conference on*. IEEE, 2005, vol. 2, pp. 25–28.

[2] Kostadin Dabov, Alessandro Foi, Vladimir Katkovnik, and Karen Egiazarian, "Image denoising by sparse 3-d transform-domain collaborative filtering," *Image Processing, IEEE Transactions on*, vol. 16, no. 8, pp. 2080–2095, 2007.

[3] Michael Elad and Michal Aharon, "Image denoising via sparse and redundant representations over learned dictionaries," *Image Processing, IEEE Transactions on*, vol. 15, no. 12, pp. 3736–3745, 2006.

[4] Alessandro Foi, Vladimir Katkovnik, and Karen Egiazarian, "Pointwise shape-adaptive dct for high-quality denoising and deblocking of grayscale and color images," *Image Processing, IEEE Transactions on*, vol. 16, no. 5, pp. 1395–1411, 2007.

[5] Julien Mairal, Francis Bach, Jean Ponce, Guillermo Sapiro, and Andrew Zisserman, "Non-local sparse models for image restoration," in *Computer Vision, 2009 IEEE 12th International Conference on*. IEEE, 2009, pp. 2272–2279.

[6] Kostadin Dabov, Alessandro Foi, and Karen Egiazarian, "Video denoising by sparse 3d transform-domain collaborative filtering," 2007.

[7] Hui Ji, Chaoqiang Liu, Zuowei Shen, and Yuhong Xu, "Robust video denoising using low rank matrix completion," in *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pp. 1791–1798.

[8] Hui Ji, Sibin Huang, Zuowei Shen, and Yuhong Xu, "Robust video restoration by joint sparse and low rank matrix approximation," *SIAM Journal on Imaging Sciences*, vol. 4, no. 4, pp. 1122–1142, 2011.

[9] Ce Liu and William T Freeman, "A high-quality video denoising algorithm based on reliable motion estimation," in *Computer Vision–ECCV 2010*, pp. 706–719. Springer, 2010.

[10] Liwei Guo, Oscar C Au, Mengyao Ma, and Zhiqin Liang, "Temporal video denoising based on multihypothesis motion compensation," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 17, no. 10, pp. 1423–1429, 2007.

[11] SM Mahbubur Rahman, M Omair Ahmad, and MNS Swamy, "Video denoising based on inter-frame statistical modeling of wavelet coefficients," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 17, no. 2, pp. 187–198, 2007.

[12] H Rabbani and S Gazor, "Video denoising in three-dimensional complex wavelet domain using a doubly stochastic modelling," *IET image processing*, vol. 6, no. 9, pp. 1262–1274, 2012.

[13] Shigong Yu, M Omair Ahmad, and MNS Swamy, "Video denoising using motion compensated 3-d wavelet transform with integrated recursive temporal filtering," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 20, no. 6, pp. 780–791, 2010.

[14] Michal Joachimiak, Dmytro Rusanovskyy, Miska M Hannuksela, and Moncef Gabbouj, "Multiview 3d video denoising in sliding 3d dct domain," in *Signal Processing Conference (EUSIPCO), 2012 Proceedings of the 20th European*. IEEE, 2012, pp. 1109–1113.

[15] Saiprasad Ravishankar and Yoram Bresler, "Learning sparsifying transforms," *Signal Processing, IEEE Transactions on*, vol. 61, no. 5, pp. 1072–1086, 2013.

[16] Saiprasad Ravishankar, Bihan Wen, and Yoram Bresler, "Online sparsifying transform learning - part 1: algorithms," *accepted to Signal Processing, IEEE Transactions on*.

[17] Saiprasad Ravishankar and Yoram Bresler, "Online sparsifying transform learning - part 2: convergence analysis," *accepted to Signal Processing, IEEE Transactions on*.

[18] Bihan Wen, Saiprasad Ravishankar, and Yoram Bresler, "Video denoising by online 3d sparsifying transform learning," in *Image Processing (ICIP), 2015 IEEE International Conference on*. IEEE, 2015, pp. 118–122.

[19] Harold C Burger, Christian J Schuler, and Stefan Harmeling, "Image denoising: Can plain neural networks compete with bm3d?," in *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*. IEEE, 2012, pp. 2392–2399.

[20] Junyuan Xie, Linli Xu, and Enhong Chen, "Image denoising and inpainting with deep neural networks," in *Advances in Neural Information Processing Systems*, 2012, pp. 341–349.

[21] Forest Agostinelli, Michael R Anderson, and Honglak Lee, "Adaptive multi-column deep neural networks with application to robust image denoising," in *Advances in Neural Information Processing Systems*, 2013, pp. 1493–1501.

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

[23] C. Qiu, N. Vaswani, B. Lois, and L. Hogben, "Recursive robust pca or recursive sparse recovery in large but structured noise," *IEEE Trans. Info. Th.*, 2014.

[24] Han Guo, Chenlu Qiu, and Namrata Vaswani, "An online algorithm for separating sparse and low-dimensional signal sequences from their sum," *Signal Processing, IEEE Transactions on*, vol. 62, no. 16, pp. 4284–4297, 2014.

[25] F. De La Torre and M. J. Black, "A framework for robust subspace learning," *International Journal of Computer Vision*, vol. 54, pp. 117–142, 2003.

[26] Jun He, Laura Balzano, and Arthur Szlam, "Incremental gradient on the grassmannian for online foreground and background separation in subsampled video," in *IEEE Conf. on Comp. Vis. Pat. Rec. (CVPR)*, 2012.

[27] Clemens Hage and Martin Kleinsteuber, "Robust pca and subspace tracking from incomplete observations using l0-surrogates," *arXiv:1210.0805 [stat.ML]*, 2013.

[28] A.E. Abdel-Hakim and M. El-Saban, "Frpca: Fast robust principal component analysis for online observations," in *Pattern Recognition (ICPR), 2012 21st International Conference on*, 2012, pp. 413–416.

[29] J. Feng, H. Xu, and S. Yan, "Online robust pca via stochastic optimization," in *Adv. Neural Info. Proc. Sys. (NIPS)*, 2013.

[30] J. Feng, H. Xu, S. Mannor, and S. Yan, "Online pca for contaminated data," in *Adv. Neural Info. Proc. Sys. (NIPS)*, 2013.

[31] Morteza Mardani, Gonzalo Mateos, and G Giannakis, "Dynamic anomalography: Tracking network anomalies via sparsity and low rank," *J. Sel. Topics in Sig. Proc.*, Feb 2013.

[32] Zihan Zhou, Xiaodong Li, John Wright, Emmanuel Candes, and Yi Ma, "Stable principal component pursuit," in *Information Theory Proceedings (ISIT), 2010 IEEE International Symposium on*. IEEE, 2010, pp. 1518–1522.