Video Denoising and Enhancement via Dynamic Sparse + Low-Rank Matrix Decomposition

Han Guo and Namrata Vaswani
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 many noisy or corrupted videos can be split into three parts - the "low-rank layer", the "sparse layer", and everything else (which is small and bounded). We show, using extensive experiments, that our denoising approach outperforms the state-of-the art denoising algorithms.**

I. 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 [2], [3], [4], [5], [6]. One of the most effective methods for image denoising, Block Matching and 3D filtering (BM3D) [3], 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 [7], the method is generalized to video denoising by searching for similar blocks across multiple frames. Other related works [8], [9] either use matrix completion [8] or sparse and low rank matrix decomposition (S+LR) [9] 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 [10], [11]; and approaches that use wavelet transform based [12], [13], [14] and discrete cosine transform (DCT) based [15] denoising solutions. Very recent video denoising methods include algorithms based on learning a sparsifying transform [16], [17], [18], [19]. Within the image denoising literature, the most promising recent approaches are based on deep learning [20], [21].

This work was partially supported by Rockwell Collins. A portion of this work was presented at IEEE Workshop on Statistical Signal Processing (SSP) 2016 [1].

Contribution. In this paper, we develop a novel approach to video denoising, called *ReProCS-based Layering Denoising (ReLD)*, that is based on the idea that many noisy or corrupted videos can be split into three parts or layers - the "low-rank layer", the "sparse layer" and the "small bounded residual layer". Here these terms mean the following. "Low-rank layer": the matrix formed by each vectorized image of this layer is low-rank. "Sparse layer": each vectorized image of this layer is sparse. "Small bounded residual layer": each vectorized image of this layer has ℓ_∞ norm that is bounded. Our proposed algorithm consists of two parts. At each time instant, it first separates the video into "noisy" versions of the two layers $\boldsymbol{\ell}_t$ and $\boldsymbol{s}_t$. This is followed by applying an existing state-of-the-art denoising algorithm, VBM3D [7] on each layer. For separating the video, we use a combination of an existing batch technique for sparse + low-rank matrix decomposition (S+LR) called principal components' pursuit (PCP) [22] and an appropriately modified version of our recently proposed dynamic robust PCA technique called Recursive Projected Compressive Sensing (ReProCS) [23], [24]. We initialize using PCP and use ReProCS afterwards to separate the video into a "sparse layer" and a "low-rank layer". The video-layering step is followed by VBM3D on each of the two layers. In doing this, VBM3D exploits the specific characteristics of each layer and, hence, is able to find more matched blocks to filter over, resulting in better denoising performance. The motivation for picking ReProCS for the layering task is its superior performance in earlier video experiments involving videos with large-sized sparse components and/or significantly changing background images.

The performance of our algorithm is compared with PCP [22], GRASTA [25] and non-convex rpca (NCRPCA) [26], which are some of the best solutions from the existing S+LR literature [22], [25], [26], [27], [28], [29], [30], [31], [32] followed by VBM3D for the denoising step; as well as with just using VBM3D directly on the video. We also compare with one image denoising method based on Multi Layer Perceptron (MLP) [33] and with [9] which proposed an approach that makes use of S+LR matrix approximation (SLMA) on grouped image patches.

A. Example applications

A large number of videos that require denoising/enhancement can be accurately modeled in the above fashion. Some examples are as

follows. All videos referenced below are posted at http://www.ece.iastate.edu/~hanguo/denoise.html.

1) In very low-light videos of moving targets/objects (the moving target is barely visible), the denoising goal is to "see" the barely visible moving targets (sparse). These are hard to see because they are corrupted by slowly-changing background images (well modeled as forming the low-rank layer plus the residual). The dark-room video is an example of this. The goal is to extract out the sparse targets or, at least, the regions occupied by these objects.
2) In a traditional denoising scenario, consider slowly changing videos that are corrupted by salt-and-pepper noise (or other impulsive noise). For these types of videos, the large magnitude part of the noise forms the "sparse layer", while the video-of-interest (slowly-changing in many applications, e.g., waterfall, moving trees, see water moving, etc) forms the approximate "low-rank layer". The approximation error in the low-rank approximation forms the "small bounded residual". See the waterfall-salt-pepper video for an example of this. The goal is to denoise or extract out the "low-rank layer".
3) More generally, consider slow-changing videos corrupted by very large variance white Gaussian noise. As we explain below, large Gaussian noise can, with high probability, be split into a very sparse noise component plus bounded noise. Thus, our approach also works on this type of videos, and in fact, in this scenario, we show that it significantly outperforms the existing state-of-the-art video denoising approaches. See the waterfall-large-noise video for an example of this.

Moreover, in all these examples, it is valid to argue that the columns of the low-rank matrix lie in a low-dimensional subspace that is either fixed or slowly changing. This is true, for example, when the background consists of moving waters, or the background changes are due to illumination variations. These also result in global (non-sparse) changes. In special cases where foreground objects are also present, the video itself become "low-rank + sparse". In such a scenario, the "sparse layer" that is extracted out will consist of the foreground object and the large magnitude part of the noise. Some examples are the curtain and lobby videos. The proposed ReLD algorithm works for these videos if VBM3D applied to the foreground layer video is able to separate out the foreground moving object(s) from the noise. As can be seen from Fig.3 that plots frame-wise peak signal to noise ratio (PSNR) for the lobby sequence, the advantage of ReLD over VBM3D is less obvious for the frames containing the foreground moving person.

B. Problem formulation

Let $\boldsymbol{m}_t$ denote the image at time t arranged as a 1D vector. We consider denoising for videos in which each image can be split as

$$\boldsymbol{m}_t = \boldsymbol{\ell}_t + \boldsymbol{s}_t + \boldsymbol{w}_t$$

where $\boldsymbol{s}_t$ is a sparse vector, $\boldsymbol{\ell}_t$'s lie in a fixed or slowly changing low-dimensional subspace so that the matrix $\boldsymbol{L} := [\boldsymbol{\ell}_1, \boldsymbol{\ell}_2, \dots, \boldsymbol{\ell}_{t_{\max}}]$ is low-rank, and $\boldsymbol{w}_t$ is the residual noise that satisfies $\|\boldsymbol{w}_t\|_\infty \le b_w$. We use $\mathcal{T}_t$ to denote the support set of $\boldsymbol{s}_t$, i.e., $\mathcal{T}_t := \mathrm{support}(\boldsymbol{s}_t)$.

In the first example given above, the moving targets' layer is $\boldsymbol{s}_t$, the slowly-changing dark background is $\boldsymbol{\ell}_t + \boldsymbol{w}_t$. The layer of interest is $\boldsymbol{s}_t$. In the second example, the slowly changing video is $\boldsymbol{\ell}_t + \boldsymbol{w}_t$, while the salt-and-pepper noise is $\boldsymbol{s}_t$. The layer of interest is $\boldsymbol{\ell}_t$. In the third example, the slowly changing video is $\boldsymbol{\ell}_t + \boldsymbol{w}_{1,t}$ with $\boldsymbol{w}_{1,t}$ being the residual; and, as we explain next, with high probability (whp), white Gaussian noise can be split as $\boldsymbol{s}_t + \boldsymbol{w}_{2,t}$ with $\boldsymbol{w}_{2,t}$ being bounded. In this case, $\boldsymbol{w}_t = \boldsymbol{w}_{1,t} + \boldsymbol{w}_{2,t}$.

Let $\boldsymbol{n}$ denote a Gaussian noise vector in $\mathbb{R}^n$ with zero mean and covariance $\sigma^2 \boldsymbol{I}$. Let $\beta(b) := 2\Phi(b) - 1$ with $\Phi(z)$ being the cumulative distribution function (CDF) of the standard Gaussian distribution. Then, it is not hard to see that $\boldsymbol{n}$ can be split as

$$\boldsymbol{n} = \boldsymbol{s} + \boldsymbol{w}$$

where $\boldsymbol{w}$ is bounded noise with $\|\boldsymbol{w}\|_\infty \le b_0$ and $\boldsymbol{s}$ is a sparse vector with support size $|\mathcal{T}_t| \approx \left(1 - \beta\left(\frac{b_0}{\sigma}\right)\right) n$ whp. More precisely, with probability at least $1 - 2\exp(-2\epsilon^2 n)$,

$$\left(1 - \beta\left(\frac{b_0}{\sigma}\right) - \epsilon\right) n \le |\mathcal{T}_t| \le \left(1 - \beta\left(\frac{b_0}{\sigma}\right) + \epsilon\right) n.$$

In words, whp, $\boldsymbol{s}$ is sparse with support size roughly $(1-\beta)n$ where $\beta = \beta\left(\frac{b_0}{\sigma}\right)$. The above claim is a direct consequence of Hoeffding's inequality for a sum of independent Bernoulli random variables[1].

II. ReProCS-based Layering Denoising (ReLD)

We summarize the ReProCS-based Layering Denoising (ReLD) algorithm in Algorithm 1 and detail each step in Algorithm 2. The approach is explained below.

Algorithm 1 Overall ReLD Algorithm

1) For $t < t_0$, initialization using PCP [22].
2) For all $t > t_0$, implement an appropriately modified ReProCS algorithm
   a) Split the video frame $\boldsymbol{m}_t$ into layers $\hat{\boldsymbol{\ell}}_t$ and $\hat{\boldsymbol{s}}_t$
   b) For every α frames, perform subspace update, i.e., update $\hat{\boldsymbol{P}}_t$
3) Denoise using VBM3D

Initialization. Take $\boldsymbol{M}_0 = [\boldsymbol{m}_1, \boldsymbol{m}_2, \cdots, \boldsymbol{m}_{t_0}]$ as training data and use PCP [22] to separate it into a sparse matrix $[\hat{\boldsymbol{s}}_1, \hat{\boldsymbol{s}}_2, \cdots, \hat{\boldsymbol{s}}_{t_0}]$ and a low-rank matrix $[\hat{\boldsymbol{\ell}}_1, \hat{\boldsymbol{\ell}}_2, \cdots, \hat{\boldsymbol{\ell}}_{t_0}]$. Compute the top $b\%$ left singular vectors of $[\hat{\boldsymbol{\ell}}_1, \hat{\boldsymbol{\ell}}_2, \cdots, \hat{\boldsymbol{\ell}}_{t_0}]$ and denote by $\hat{\boldsymbol{P}}_0$. Here $b\%$ left singular vectors of a matrix $\boldsymbol{M}$

[1] If p is the probability of $z_i = 1$, then Hoeffding's inequality says that:

$$\Pr((p-\epsilon)n \le \sum_i z_i \le (p+\epsilon)n) \ge 1 - 2\exp(-2\varepsilon^2 n)$$

We apply it to the Bernoulli random variables z_i's with z_i defined as $z_i = 1$ if $\{\boldsymbol{s}_i \ne 0\}$ and $z_i = 0$ if $\{\boldsymbol{s}_i = 0\}$. Clearly, $\Pr(z_i = 0) = \Pr(\boldsymbol{s}_i = 0) = \Pr(\boldsymbol{n}_i^2 \le b_0^2) = \Phi(b_0/\sigma) - \Phi(-b_0/\sigma) = 2\Phi(b_0/\sigma) - 1 = \beta(b_0/\sigma)$.

Algorithm 2 Details of each step of ReLD

Parameters: We used $\alpha = 20, K_{\min} = 3, K_{\max} = 10, t_0 = 50$ in all experiments.

1) Initialization using PCP [22]: Compute $(\hat{\boldsymbol{L}}_0, \hat{\boldsymbol{S}}_0) \leftarrow \text{PCP}(\boldsymbol{M}_0)$ and compute $[\hat{\boldsymbol{P}}_0, \hat{\boldsymbol{\Sigma}}_0] \leftarrow \text{approx-basis}(\hat{\boldsymbol{L}}_0, 90\%)$. The notation PCP($\boldsymbol{M}$) means implementing the PCP algorithm on matrix $\boldsymbol{M}$ and $\boldsymbol{P}$ = approx-basis($\boldsymbol{M}, b\%$) means that $\boldsymbol{P}$ is the $b\%$ left singular vectors' matrix for $\boldsymbol{M}$.
   Set $\hat{r} \leftarrow \text{rank}(\hat{\boldsymbol{P}}_0)$, $\hat{\boldsymbol{\sigma}}_{\min} \leftarrow (\hat{\boldsymbol{\Sigma}}_0)_{\hat{r},\hat{r}}$, $\hat{t}_0 = t_0$, flag=detect
2) For all $t > t_0$, implement an appropriately modified ReProCS algorithm
   a) Split $\boldsymbol{m}_t$ into layers $\hat{\boldsymbol{\ell}}_t$ and $\hat{\boldsymbol{s}}_t$:
      i) Compute $\boldsymbol{y}_t \leftarrow \boldsymbol{\Phi}_t \boldsymbol{m}_t$ with $\boldsymbol{\Phi}_t \leftarrow \boldsymbol{I} - \hat{\boldsymbol{P}}_{t-1}\hat{\boldsymbol{P}}_{t-1}'$
      ii) Compute $\hat{\boldsymbol{s}}_t$ as the solution of
      $$\min_{\boldsymbol{x}} \|\boldsymbol{x}\|_1 \text{s.t.} \|\boldsymbol{y}_t - \boldsymbol{\Phi}_t \boldsymbol{x}\|_2 \le \xi$$
      with $\xi = \|\boldsymbol{\Phi}_t \hat{\boldsymbol{\ell}}_{t-1}\|$
      iii) $\hat{\mathcal{T}}_t \leftarrow \text{Thresh}(\hat{\boldsymbol{s}}_t, \omega)$ with $\omega = 3\sqrt{\|\boldsymbol{m}_t\|^2/n}$. Here $\mathcal{T} \leftarrow \text{Thresh}(\boldsymbol{x}, \omega)$ means that $\mathcal{T} = \{i : |(\boldsymbol{x})_i| \ge \omega\}$
      $\hat{\boldsymbol{s}}_{t,*} \leftarrow \text{LS}(\boldsymbol{y}_t, \boldsymbol{\Phi}_t, \hat{\mathcal{T}}_t)$. Here $\hat{\boldsymbol{x}} \leftarrow \text{LS}(\boldsymbol{y}, \boldsymbol{A}, \mathcal{T})$ means that $\hat{\boldsymbol{x}}_{\mathcal{T}} = (\boldsymbol{A}_{\mathcal{T}}'\boldsymbol{A}_{\mathcal{T}})^{-1}\boldsymbol{A}_{\mathcal{T}}'\boldsymbol{y}$, which is least-squared estimate of $\boldsymbol{x}$ on $\mathcal{T}$.
      iv) $\hat{\boldsymbol{\ell}}_t \leftarrow \boldsymbol{m}_t - \hat{\boldsymbol{s}}_t$, $\hat{\boldsymbol{\ell}}_{t,*} \leftarrow \boldsymbol{m}_t - \hat{\boldsymbol{s}}_{t,*}$
   b) Perform subspace update, i.e., update $\hat{\boldsymbol{P}}_t$ (see details in supplementary material)
3) Denoise using VBM3D:
   a) $\hat{\sigma}_{\text{fg}} \leftarrow \text{Std-est}([\hat{\boldsymbol{s}}_t, \dots, \hat{\boldsymbol{s}}_{t_0}])$
   $\hat{\sigma}_{\text{bg}} \leftarrow \text{Std-est}([\hat{\boldsymbol{\ell}}_t, \dots, \hat{\boldsymbol{\ell}}_{t_0}])$. Here Std-est($\boldsymbol{M}$) denotes estimating the standard deviation of noise from $\boldsymbol{M}$: we first subtract column-wise mean from $\boldsymbol{M}$ and then compute the standard deviation by seeing it as a vector.
   b) $\hat{\boldsymbol{S}}_{\text{denoised}} \leftarrow \text{VBM3D}([\hat{\boldsymbol{s}}_1, \dots, \hat{\boldsymbol{s}}_{t_{\max}}], \hat{\sigma}_{\text{fg}})$
   $\hat{\boldsymbol{L}}_{\text{denoised}} \leftarrow \text{VBM3D}([\hat{\boldsymbol{\ell}}_1, \dots, \hat{\boldsymbol{\ell}}_{t_{\max}}], \hat{\sigma}_{\text{bg}})$. Here VBM3D($\boldsymbol{M}, \sigma$) implements the VBM3D algorithm on matrix $\boldsymbol{M}$ with input standard deviation σ.

Output: $\hat{\boldsymbol{S}}_{\text{denoised}}$, $\hat{\boldsymbol{L}}_{\text{denoised}}$ or $\hat{\boldsymbol{I}}_{\text{denoised}} = \hat{\boldsymbol{S}}_{\text{denoised}} + \hat{\boldsymbol{L}}_{\text{denoised}}$ based on applications

refer to the left singular vectors of $\boldsymbol{M}$ whose corresponding singular values form the smallest set of singular values that contains at least $b\%$ of the total singular values' energy.

Splitting phase. Let $\hat{\boldsymbol{P}}_{t-1}$ be the basis matrix (matrix with orthonormal columns) for the estimated subspace of $\boldsymbol{\ell}_{t-1}$. For $t \ge t_0+1$, we split $\boldsymbol{m}_t$ into $\hat{\boldsymbol{s}}_t$ and $\hat{\boldsymbol{\ell}}_t$ using prac-ReProCS [24]. To do this, we first project $\boldsymbol{m}_t$ onto the subspace orthogonal to range($\hat{\boldsymbol{P}}_{t-1}$) to get the projected measurement vector,

$$\boldsymbol{y}_t := (\boldsymbol{I} - \hat{\boldsymbol{P}}_{t-1}\hat{\boldsymbol{P}}_{t-1}')\boldsymbol{m}_t := \boldsymbol{\Phi}_t \boldsymbol{m}_t. \quad (1)$$

Observe that $\boldsymbol{y}_t$ can be expressed as

$$\boldsymbol{y_t} = \boldsymbol{\Phi}_t \boldsymbol{s}_t + \boldsymbol{\beta}_t \text{ where } \boldsymbol{\beta}_t := \boldsymbol{\Phi}_t(\boldsymbol{\ell}_t + \boldsymbol{w}_t). \quad (2)$$

Because of the slow subspace change assumption, the projection nullifies most of the contribution of $\boldsymbol{\ell}_t$ and hence $\boldsymbol{\beta}_t$ is small noise. The problem of recovering $\boldsymbol{s}_t$ from $\boldsymbol{y}_t$ becomes a traditional noisy sparse recovery/CS problem and one can use $\boldsymbol{\ell}_1$ minimization or any of the greedy or iterative thresholding algorithms to solve it. We denote its solution by $\hat{\boldsymbol{s}}_t$, and obtain $\hat{\boldsymbol{\ell}}_t$ by simply subtracting $\hat{\boldsymbol{s}}_t$ from $\boldsymbol{m}_t$.

Denoising phase. We perform VBM3D on $\hat{\boldsymbol{s}}_t$'s and $\hat{\boldsymbol{\ell}}_t$'s and obtain the denoised data $\hat{\boldsymbol{S}}_{\text{denoised}}$ and $\hat{\boldsymbol{L}}_{\text{denoised}}$. Based on applications, we output different results. For example, in the low-light denoising case, our output is $\hat{\boldsymbol{S}}_{\text{denoised}}$ since the goal is to extract out the sparse targets. In traditional denoising scenarios, the output can be $\hat{\boldsymbol{L}}_{\text{denoised}}$ or $\hat{\boldsymbol{I}}_{\text{denoised}}$=$\hat{\boldsymbol{S}}_{\text{denoised}}$+$\hat{\boldsymbol{L}}_{\text{denoised}}$. This depends on whether the video contains only background or background & foreground. In practice, even for videos with only backgrounds, adding $\hat{\boldsymbol{S}}_{\text{denoised}}$ helps improve PSNR.

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

III. Experiments

Due to limited space, we only present some experimental results in this paper. The complete presentation of experimental results are in the supplementary material.

A. Removing Additive Noise

First, we compare the performance of our propsed denoising framework with the separating algorithm being ReProCS, PCP [22], NCRPCA [26], and GRASTA [25] (We call them ReLD, PCP-LD, NCRPCA-LD, and GRASTA-LD for short). We test the algorithms on the Waterfall dataset (downloaded from Youtube https://www.youtube.com/watch?v=UwSzu_0h7Bg). Besides these Laying-Denoising algorithms, we also compare with VBM3D and a neural network image denoising method, Multi Layer Perceptron (MLP) [33]. The codes for algorithms being compared are downloaded from the authors' webpages. 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. The video is a background scene without foreground, and can be well modeled as low-rank. We add i.i.d. Gaussian noise with different variance onto the video.

The video consists of 650 frames and the images are of size 1080×1920. To speed up the algorithms, we test on the under-sampled data which reduce the image size to 108×192. As can be seen from TABLE I, ReLD has the best denoising performance.

Next we thoroughly compare the denoising performance on two more dataset – curtain and lobby which are available at http://www.ece.iastate.edu/~hanguo/denoise.html. The algorithms being compared are ReLD, VBM3D, MLP and SLMA. The noise being added to the original image frames

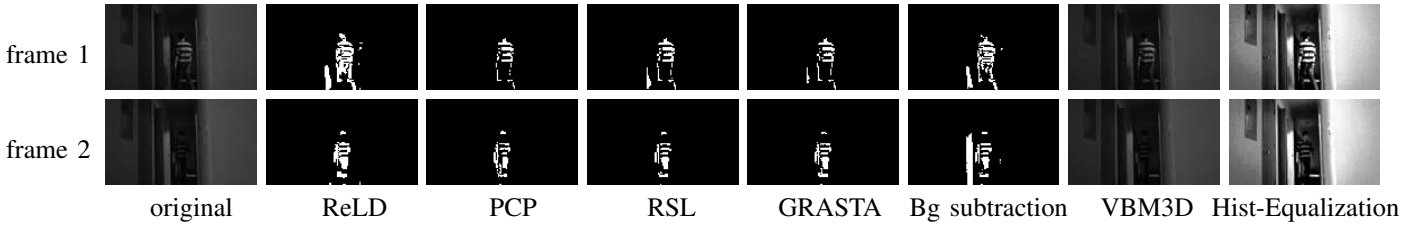

Fig. 1: Ability of "seeing" in the dark for three sample frames. From left to right: orignal dark image, results by ReLD, PCP [22], RSL [27], GRASTA[25], background subtraction, VBM3D and Histogram-Equalization.

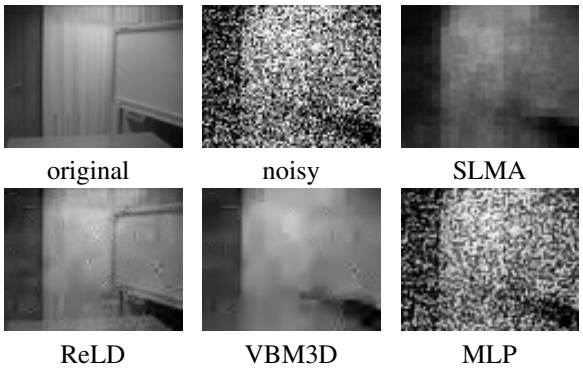

Fig. 2: Visual comparison of denoising performance for the Curtain dataset for very large Gaussian noise ($\sigma = 70$)

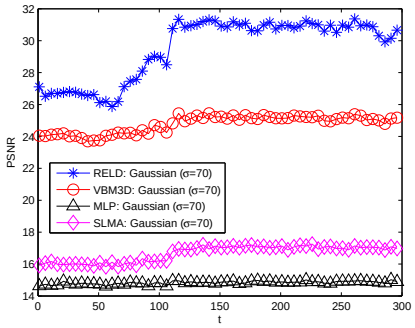

Fig. 3: Frame-wise PSNR for Lobby dataset with different noise level: Gaussian noise with $\sigma = 70$.

are Gaussian ($\sigma = 70$). In Fig.2 we show sample visual comparisons of the denoising performance for Curtain dataset and in Fig.3 we compute the frame-wise PSNRs for Lobby dataset. Both figure show that ReLD outperforms all other algorithms – the PSNR is the highest in all image frames, and visually, ReLD is able to recover more details of the images while other algorithms either fail or cause severe blurring effect.

We compare on two more dataset and summarize all results in Table I.

B. 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 barely visible person walked through the hallway. Note that the video is captured in a low-light setting where most of the details can be observed by implementing histogram equalization (shown in Fig. 1, last column). The algorithms being compared are foreground/background separation techniques – PCP [22], RSL [27], GRASTA [25], background subtraction, and video denoising approach VBM3D. The output we are using here is $\hat{\boldsymbol{S}}_{\text{denoised}}$ and only its support (properly thresholded) is shown for ease of display. Fig. 1 shows that VBM3D is not capable of observing the walking person while our algorithm can successfully mark the interested foreground person and the result is slightly better than or as good as other foreground/background separation techniques.

σ	Dataset: waterfall			
	ReLD	VBM3D	MLP	PCP-LD
50	33.08(73.14)	27.99(24.14)	18.87(477.60)	32.93(195.77)
70	29.25(69.77)	24.42(21.01)	15.03(478.73)	29.17(197.94)
σ	NCRPCA-LD		GRASTA-LD	
50	30.48(128.35)		25.33(58.23)	
70	27.97(133.53)		21.89(55.45)	
σ	Dataset: fountain			
	ReLD	VBM3D	MLP	SLMA
50	30.53(15.82)	26.55(5.24)	18.53(109.79)	18.55(3.13×10^4)
70	27.53(15.03)	22.08(4.69)	14.85(107.52)	16.25(3.19×10^4)
σ	Dataset: escalator			
	ReLD	VBM3D	MLP	SLMA
50	27.84(16.03)	25.10(5.27)	18.83 (109.40)	17.98(3.21×10^4)
70	25.15(15.28)	20.20(4.72)	15.20(108.78)	15.90(3.18×10^4)
σ	Dataset: curtain			
	ReLD	VBM3D	MLP	SLMA
50	31.91(17.17)	30.29(4.42)	18.58(188.30)	19.12(7.86×10^4)
70	28.10(16.50)	26.15(3.85)	14.73(192.00)	16.68(8.30×10^4)
σ	Dataset: lobby			
	ReLD	VBM3D	MLP	SLMA
50	35.15(58.41)	29.23(19.35)	18.66(403.59)	18.21(3.99×10^5)
70	29.68(56.51)	24.90(17.00)	14.85(401.29)	16.82(4.09×10^5)

TABLE I: PSNR (and running time in second) for different denoising algorithms.

IV. 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. In general, splitting the video first results in a clean low-rank layer since the large noise goes to the sparse layer in the ℓ_1 minimization (2(a)ii). The clean layer improves the "grouping" accuracy in VBM3D. One drawback of our algorithm is that VBM3D need to be executed due to the

separated two layers. As a result, the running time is at least doubled. We reserve the exploit for more efficient combination of VBM3D and matrix decomposition for our future work.

REFERENCES

[1] Han Guo and Namrata Vaswani, "Video denoising via online sparse and low-rank matrix decomposition," in *Statistical Signal Processing Workshop (SSP), 2016 IEEE*. IEEE, 2016, pp. 1–5.

[2] 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.

[3] 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.

[4] 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.

[5] 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.

[6] 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.

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

[8] 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.

[9] 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.

[10] 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.

[11] 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.

[12] 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.

[13] 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.

[14] 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.

[15] 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.

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

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

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

[19] 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.

[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] 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.

[26] Praneeth Netrapalli, UN Niranjan, Sujay Sanghavi, Animashree Anandkumar, and Prateek Jain, "Non-convex robust pca," in *Advances in Neural Information Processing Systems*, 2014, pp. 1107–1115.

[27] 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.

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

[29] 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.

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

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

[32] 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.

[33] 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.