VIDEO DENOISING BY ONLINE 3D SPARSIFYING TRANSFORM LEARNING

Bihan Wen, Saiprasad Ravishankar, and Yoram Bresler

Department of Electrical and Computer Engineering and the Coordinated Science Laboratory,
University of Illinois at Urbana-Champaign, IL, USA

ABSTRACT

Exploiting the sparsity of signals in an adaptive dictionary or transform domain benefits various applications in image/video processing. As opposed to synthesis dictionary learning, transform learning allows for cheap computations, and has been demonstrated to perform well in applications such as image denoising. Very recently, we proposed methods for online sparsifying transform learning, which are particularly useful for processing large-scale or streaming data. Online transform learning has good convergence guarantees and enjoys a much lower computational cost than online synthesis dictionary learning. In this work, we present a video denoising framework based on online 3D spatio-temporal sparsifying transform learning. The proposed scheme has low computational and memory costs, and can potentially handle streaming video. Our numerical experiments show promising performance for the proposed video denoising method compared to popular prior or state-of-the-art methods.

***Index Terms*—** Sparsifying transforms, Denoising, Online learning, Sparse representations, Big data.

1. INTRODUCTION

Denoising is one of the most fundamental problems in signal processing. The goal in denoising is to take corrupted signals, images or video and process them to obtain clean or high-quality estimates. This is especially useful for applications that require high-quality signals and images such as medical imaging applications, surveillance video, etc. The ubiquitous use of relatively low-quality smart phone cameras has also led to the increasing importance of video denoising.

Several methods have been proposed in the past for the denoising of video data. Some of these methods are based on motion estimation and compensation [1, 2]. In these methods, on top of spatial similarity, temporal redundancy is exploited by filtering along the estimated motion trajectories. Other video denoising methods exploit the sparsity of video data in some known transform domain or dictionary such as the discrete cosine transform (DCT), or Wavelets, to enable better noise attenuation [3, 4]. Non-local methods have also become very popular in video denoising in recent years. Methods such as VBM3D [5] and VBM4D [6] have been shown to provide excellent performance in video denoising. These methods also exploit sparsifying transforms such as the DCT as part of their framework.

Recently, the adaptation of sparse models (such as the synthesis dictionary model [7, 8], analysis dictionary model [9], or transform model [10, 11]) based on training signals has received increasing attention [9, 10, 12–17], and has been shown to be beneficial in various applications including image or video denoising. While the data-driven adaptation of synthesis dictionaries for the purpose of denoising video or 3D data [18, 19] has been studied in some recent papers, the usefulness of learned sparsifying transforms has not been explored in these applications.

In this work, we focus on video denoising and propose a novel framework based on learned 3D sparsifying transforms. As opposed to the synthesis dictionary model, where sparse coding is NP-hard (Non-deterministic Polynomial-time hard) [20, 21], the transform model has the advantage that sparse coding in the model can be perfomed exactly and cheaply by zeroing out all but a certain number of non-zero transform coefficients of largest magnitude. The learning of sparsifying transforms is typically much cheaper than synthesis, or analysis dictionary learning [10, 22]. Very recently, we introduced the idea of online learning of sparsiying transforms for signals or image patches [23, 24]. Online learning is particularly useful for big data, and for applications such as real-time denoising, i.e., denoising of streaming data. As opposed to batch transform learning [10], where the transform is learnt using all the training data simultaneously, online transform learning has the advantage that it handles (training) data sequentially, and involves much cheaper computations, and lower latency and memory requirements. It has also been shown to be cheaper than online overcomplete synthesis dictionary learning [23].

While we have shown the usefulness of online transform learning for large-scale image denoising [23], the usefulness of transform learning (either online or batch) for video denoising has not been explored. Moreover, video data typically have redundancy along the time axis, which will not be captured by learning sparsifying transforms for the 2D patches of the video frames. Therefore, in this paper, we propose a novel online video denoising scheme based on 3D sparsifying transform learning. Our framework iteratively adapts the sparsifying transform and sparse codes for (overlapping) 3D (spatio-temporal) patches that are extracted sequentially from groups of frames. Denoised versions of the 3D patches are estimated in each iteration of our algorithm, and denoised versions of the video frames are estimated by averaging the denoised 3D patches at their respective spatio-temporal locations. Our numerical results demonstrate the promising performance of the proposed method as compared to well-known alternatives such as adaptive overcomplete synthesis dictionary-based denoising [19], 3D DCT-based denoising, or non-local methods including VBM3D [5], and VBM4D [6].

2. DENOISING PROBLEM FORMULATIONS

We briefly discuss the recently proposed formulations for denoising based on online and mini-batch transform learning [23, 25].

2.1. Signal or Image Denoising by Online Transform Learning

The goal in denoising is to recover an estimate of a signal u from the measurement $y = u + e$, corrupted by additive noise e. Here, we consider a time sequence of measurements $\{y_t\}$, with $y_t = u_t + e_t$,

This work was supported in part by the National Science Foundation (NSF) under grant CCF-1320953.

and $e_t \in \mathbb{R}^n$ being the noise. We assume e_t whose entries are independent and identically distributed (i.i.d.) Gaussian with zero mean and variance σ_t^2. The goal of online denoising is to recover estimates of $u_t \; \forall \; t$. We model the underlying signals as approximately sparse in an (unknown) transform domain.

In prior work [23], we proposed a denoising methodology based on online sparsifying transform learning, where the transform is adapted based on sequentially processed data. For time $t = 1, 2, 3, ...$, the problem of updating the adaptive transform and sparse code (i.e., the sparse representation in the adaptive transform domain) to account for the new noisy signal $y_t \in \mathbb{R}^n$ is

$$(\text{P1}) \left\{ \hat{W}_t, \hat{x}_t \right\} = \underset{W,\, x_t}{\arg\min}\; \frac{1}{t} \sum_{\tau=1}^{t} \left\{ \|W y_\tau - x_\tau\|_2^2 + \lambda_\tau \nu(W) \right\}$$

$$+ \frac{1}{t} \sum_{\tau=1}^{t} \alpha_\tau^2 \left\| x_\tau \right\|_0 \qquad s.t.\;\; x_\tau = \hat{x}_\tau,\; 1 \leq \tau \leq t-1$$

where $\nu(W) = -\log |\det W| + \|W\|_F^2$ is a transform learning regularizer [10], $\lambda_\tau = \lambda_0 \|y_\tau\|_2^2$ with $\lambda_0 > 0$, and the weights $\alpha_\tau \propto \sigma_\tau$. The $\|\cdot\|_0$ operation counts the number of non-zeros in a vector or matrix. Matrix $\hat{W}_t$ in (P1) is the optimal transform at time t, and $\hat{x}_t$ is the optimal sparse code for y_t. Note that at time t, only the latest optimal sparse code $\hat{x}_t$ is updated in (P1) [1] along with the transform $\hat{W}_t$. The condition $x_\tau = \hat{x}_\tau,\; 1 \leq \tau \leq t-1$, is therefore assumed. For brevity, we will not explicitly restate this condition (or, its appropriate variant) in the formulations in the rest of this paper.

The regularizer $\nu(W)$ in (P1) prevents trivial solutions and controls the condition number and scaling of the learnt transform [10]. The condition number $\kappa(W)$ is upper bounded by a monotonically increasing function of $\nu(W)$. In the limit $\lambda_0 \to \infty$ (and assuming the y_τ, $1 \leq \tau \leq t$, are not all zero), the condition number of the optimal transform in (P1) tends to 1, and its spectral norm tends to $1/\sqrt{2}$. In practice, the transforms learnt via (P1) are well conditioned for finite λ_0 [23]. The specific choice of λ_0 (i.e., condition number) depends on the application.

A simple least-squares denoised signal estimate is obtained using (P1) at each time t as $\hat{u}_t = \hat{W}_t^{-1} \hat{x}_t$. Problem (P1) can also be used for patch-based denoising of large images [23]. The overlapping patches of the noisy images are processed sequentially, and the denoised image is obtained by averaging the denoised patches at their respective image locations.

For non-stationary data, it may not be desirable to uniformly fit a single transform W to all the y_τ, $1 \leq \tau \leq t$, in (P1). We previously proposed [23] to address this case by introducing a forgetting factor $\rho^{t-\tau}$ (with a constant $0 < \rho < 1$), that scales the terms in (P1). Such a forgetting factor diminishes the influence of "old" data. The objective function in (P1) is then modified as

$$\frac{1}{t} \sum_{\tau=1}^{t} \rho^{t-\tau} \left\{ \|W y_\tau - x_\tau\|_2^2 + \lambda_\tau \nu(W) + \alpha_\tau^2 \left\| x_\tau \right\|_0 \right\} \qquad (1)$$

Another useful variation of Problem (P1) involves *mini-batch* learning, where a block, or group, or mini-batch of signals is processed at a time [23]. Assuming a fixed block size M, the L^{th} ($L \geq 1$) block of signals is $Y_L = \left[y_{LM-M+1} \mid y_{LM-M+2} \mid \;\; ... \;\; \mid y_{LM} \right]$. For $L = 1, 2, 3, ...$, the mini-batch sparsifying transform learning problem is

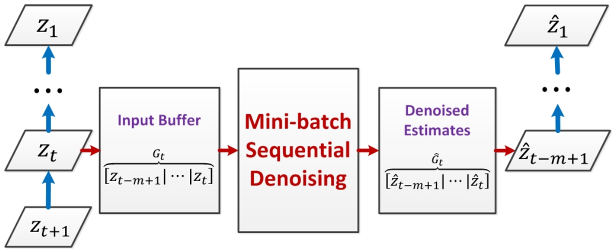
z_1
z_t
z_{t+1}
Input Buffer
G_t
$[z_{t-m+1}| \cdots |z_t]$
Mini-batch Sequential Denoising
Denoised Estimates
$\hat{G}_t$
$[\hat{z}_{t-m+1}| \cdots |\hat{z}_t]$
$\hat{z}_{t-m+1}$
$\hat{z}_1$

Fig. 1. A simple illustration of the proposed online video denoising scheme by 3D sparsifying transform learning.

$$\left\{ \hat{W}_L, \hat{X}_L \right\} = \underset{W,\, X_L}{\arg\min}\; \frac{1}{LM} \sum_{j=1}^{L} \left\{ \|W Y_j - X_j\|_F^2 + \Lambda_j \, \nu(W) \right\}$$

$$+ \frac{1}{LM} \sum_{j=1}^{L} \sum_{i=1}^{M} \alpha_{jM-M+i}^2 \left\| x_{jM-M+i} \right\|_0 \qquad (\text{P2})$$

where the regularizer weight is $\Lambda_j = \lambda_0 \|Y_j\|_F^2$, and the matrix $X_L = \left[x_{LM-M+1} \mid x_{LM-M+2} \mid \;\; ... \;\; \mid x_{LM} \right]$ contains the block of sparse codes corresponding to Y_L. A simple denoised estimate of the noisy block of signals in Y_L is obtained for each L as $\hat{U}_L = \hat{W}_L^{-1} \hat{X}_L$. The mini-batch transform learning Problem (P2) is a generalized version of (P1), with (P2) being equivalent to (P1) for $M = 1$. Mini-batch learning can provide potential speedups over the $M = 1$ case in applications, but this comes at the cost of higher memory requirements and latency [23].

2.2. Online Video Denoising Framework

Prior work on adaptive sparsifying transform-based image denoising [22, 23, 26] learnt the transform matrix from 2D image patches. However, in video denoising, exploiting the sparsity and redundancy in both the spatial and temporal dimensions typically leads to better performance than denoising each frame separately [18, 19]. We therefore propose online video denoising by sparsifying transform learning on 3D spatio-temporal patches.

Fig. 1 illustrates the framework of our proposed online video denoising scheme. The frames of the noisy video (assumed to be corrupted by additive i.i.d. Gaussian noise) denoted as $z_\tau \in \mathbb{R}^{a \times b}$ arrive at $\tau = 1, 2, 3$, etc. At time $\tau = t$, the newly arrived frame z_t is added to a fixed-size FIFO (first in first out) buffer that stores a block of m consecutive frames $\{z_i\}_{i=t-m+1}^{t}$, and the oldest frame z_{t-m} is dropped. We denote this spatio-temporal tensor data of frames stacked along the temporal dimension as $G_t = \left[z_{t-m+1} \mid z_{t-m+2} \mid \;\; ... \;\; \mid z_t \right]$, with $G_t \in \mathbb{R}^{a \times b \times m}$. (For $t < m$, the initial frames in G_t are set to all-zero frames.) The partially overlapping $n_1 \times n_2 \times n_3$ size 3D patches of G_t are extracted sequentially in a spatially and temporally contiguous order. A spatio-temporal sparsifying transform is adapted in an online manner, and used to denoise the 3D patches. Each noisy frame z_t arises once in each of the tensors in the set $\{G_j\}_{j=t}^{t+m-1}$. Thus, the denoised estimate (output) of each z_t is computed by averaging the corresponding denoised overlapping 3D patches from the m overlapping tensors at their respective 3D locations. As a result, there is (at least) an $m-1$ frame delay between the arrival of z_t and the generation of its final denoised estimate. In Fig. 1, $\hat{G}_t$ stores the most up-to-date denoised estimate of each noisy frame (obtained

[1] This is because only the signal y_t is assumed to be stored in memory at time t for the online scheme.

by averaging the denoised 3D patches from the overlapping tensors processed so far) in G_t. Only the leftmost frame $\hat{z}_{t-m+1}$ in $\hat{G}_t$ is output at time t, since all other frame estimates will be updated further based on future G_τ's ($\tau > t$).

We now discuss the formulation for sequentially denoising the 3D spatio-temporal patches in each G_t (for $t = 1, 2, 3$, etc.). Let $R_i G_t \in \mathbb{R}^n$ (with $n = n_1 n_2 n_3$, $n_3 \leq m$) denote the vectorized form of the i$^{\text{th}}$ 3D patch extracted from G_t (a total of P partially overlapping patches are assumed for each G_t), with R_i being a patch-extraction operator. We process a group, or mini-batch, of M 3D patches at a time from G_t in 3D frame-major raster scan order. To impose spatio-temporal contiguity of 3D patches from two adjacent tensors, for each t, we reverse the raster scan order between G_t and G_{t+1}. Let N be the total number of mini-batches in each G_t ($N = P/M$). Then, for a particular time t, we solve the following transform learning problem for each $k = 1, 2, 3, ..., N$, to adapt the transform and sparse codes based on the k$^{\text{th}}$ mini-batch in G_t

$$\left\{\hat{W}_{L_k}, \hat{X}_{L_k}\right\} = \underset{W, X_{L_k}}{\arg\min} \frac{1}{L_k M} \sum_{j=1}^{L_k} \rho^{L_k - j} \left\{ \|WY_j - X_j\|_F^2 \right\}$$

$$+ \frac{1}{L_k M} \sum_{j=1}^{L_k} \rho^{L_k - j} \left\{ \Lambda_j \, \nu(W) + \sum_{i=1}^{M} \alpha_{j,i}^2 \, \|X_{j,i}\|_0 \right\} \quad \text{(P3)}$$

where $L_k \triangleq N \times (t-1) + k$. In (P3), the matrix $Y_j = \left[R_{lM-M+1} G_{q+1} \mid R_{lM-M+2} G_{q+1} \mid ... \mid R_{lM} G_{q+1} \right] \in \mathbb{R}^{n \times M}$, with $q \triangleq \lfloor j/N \rfloor$ and $l \triangleq j - qN$, indexes the mini-batches processed from the various G_τ, $1 \leq \tau \leq t$. The matrix $X_j \in \mathbb{R}^{n \times M}$ denotes the sparse codes corresponding to the mini-batch Y_j, and $X_{j,i}$ denotes the i$^{\text{th}}$ column of X_j, whose sparsity weight in (P3) is $\alpha_{j,i}^2$. Note that the factor $L_k M$ in (P3) denotes the total number of 3D patches processed from all G_τ, $1 \leq \tau \leq t$. We use a forgetting factor $\rho^{L_k - j}$ in (P3) to diminish the influence of old frames or old mini-batches. Once (P3) is solved, the denoised version of the current mini-batch of noisy signals is computed simply as $\hat{U}_{L_k} = \hat{W}_{L_k}^{-1} \hat{X}_{L_k}$.

3. ALGORITHMS AND PROPERTIES

We refer to our video denoising methodology by solving (P3) as VIDOLSAT (VIdeo Denoising by Online Learning of SpArsifying Transforms). Our proposed method for (P3) involves a sparse coding step and a transform update step [23]. This is followed by another sparse coding step to improve the accuracy of the solution.

3.1. Sparse Coding

In the sparse coding step, we solve for $\hat{X}_{L_k}$ in (P3) with fixed $W = \hat{W}_{L_k - 1}$, as follows

$$\hat{X}_{L_k} = \underset{X_{L_k}}{\arg\min} \, \|WY_{L_k} - X_{L_k}\|_F^2 + \sum_{i=1}^{M} \alpha_{L_k,i}^2 \, \|X_{L_k,i}\|_0 \quad (2)$$

A solution in (2) is given as $\hat{X}_{L_k,i} = \hat{H}_{\alpha_{L_k,i}}(WY_{L_k,i}) \; \forall \; i$ [23]. Here, the hard thresholding operator $\hat{H}_\alpha(\cdot)$ is defined as

$$\left(\hat{H}_\alpha(b)\right)_p = \begin{cases} 0 & , \; |b_p| < \alpha \\ b_p & , \; |b_p| \geq \alpha \end{cases} \quad (3)$$

where $b \in \mathbb{R}^n$, and the subscript p indexes vector entries. This simple hard thresholding operation for sparse coding is similar to traditional techniques involving analytical sparsifying transforms [27].

3.2. Transform Update

In the transform update step, we solve Problem (P3) for W with fixed $X_j = \hat{X}_j$, $1 \leq j \leq L_k$, as follows

$$\min_W \frac{1}{L_k M} \sum_{j=1}^{L_k} \rho^{L_k - j} \left\{ \|WY_j - X_j\|_F^2 + \Lambda_j \nu(W) \right\} \quad (4)$$

This problem has a closed-form solution (similar to Section III-B2 in [23]).

Define $b_k = L_k M$. Let $P_{L_k} \in \mathbb{R}^{n \times n}$ be the square root of $b_k^{-1} \sum_{j=1}^{L_k} \rho^{L_k - j} (Y_j Y_j^T + \Lambda_j I)$. Denoting the full singular value decomposition (SVD) of $P_{L_k}^{-1} \Theta_{L_k}$ as $Q_{L_k} \Sigma_{L_k} U_{L_k}^T$, with $\Theta_{L_k} = b_k^{-1} \sum_{j=1}^{L_k} \rho^{L_k - j} Y_j X_j^T$, we then have that the closed-form solution to (4) is

$$\hat{W}_{L_k} = 0.5 U_{L_k} \left(\Sigma_{L_k} + \left(\Sigma_{L_k}^2 + 2\beta_{L_k} I \right)^{\frac{1}{2}} \right) Q_{L_k}^T P_{L_k}^{-1} \quad (5)$$

where I denotes the identity matrix, and $(\cdot)^{\frac{1}{2}}$ denotes the positive definite square of a positive definite matrix. The quantities $\Gamma_{L_k} \triangleq b_k^{-1} \sum_{j=1}^{L_k} \rho^{L_k - j} Y_j Y_j^T$, Θ_{L_k}, and $\beta_{L_k} \triangleq \sum_{j=1}^{L_k} b_k^{-1} \Lambda_j$ are all computed sequentially over time [23].

3.3. Multi-pass Denoising

In order to further enhance the denoising performance, we perform multiple passes of denoising for each G_t in our framework [26]. In each pass, we construct the Y_j's in (P3) using the 3D patches extracted from the denoised estimates of the G_t's from the previous pass. As the sparsity penalty weights are set proportional to the noise level, $\alpha_{j,i} \propto \sigma$, the noise level σ in each such pass is set to an estimate of the remaining noise in the denoised G_t's from the previous pass.

3.4. VIDOLSAT Properties

The per-frame computational cost of the proposed VIDOLSAT algorithm is $O(n^2 PK)$, where $W \in \mathbb{R}^{n \times n}$, P is the number of partially overlapping patches in G_t, and K is the number of passes in the multi-pass scheme. Assuming $J \gg nM/m$ (large videos), the proposed algorithms have memory cost scaling as $O(Jm)$, where m is the number of frames in G_t, and J is the number of pixels in each frame.

4. NUMERICAL EXPERIMENTS

In this section, we present preliminary results for our VIDOLSAT algorithm [2]. We work with the standard gray-scale videos *Salesman* (288 × 352 × 50), *Miss America* (288 × 360 × 150), and *Coastguard* (144 × 176 × 300) (available at [28]), and simulate i.i.d. Gaussian noise at 5 different noise levels ($\sigma = 5, 10, 15, 20, 50$) for each video. We compare the denoising results obtained by our VIDOLSAT algorithm to those obtained by popular methods such as VBM3D [5], VBM4D [6], sparse K-SVD denoising [19], and our own patch-based 3D DCT denoising (same as the VIDOLSAT method, but uses 3D DCT instead of the learned transform). We used the publicly available implementations of the sparse K-SVD [29], VBM3D and VBM4D [28] algorithms.

[2] A Matlab implementation of VIDOLSAT that can reproduce these results is publicly available at http://www.ifp.illinois.edu/~yoram.

σ	*Salesman*		*Miss America*		*Coastguard*	
5	40.87	41.03	42.03	41.99	38.47	38.55
	40.43	40.82	41.51	41.88	38.32	39.12
	41.55	**41.69**	42.30	**42.33**	**39.60**	39.53
10	36.92	37.02	39.46	39.72	34.61	34.75
	37.29	37.12	39.64	39.85	34.82	35.35
	37.84	**38.02**	40.31	**40.34**	**35.73**	35.67
15	34.66	34.73	37.66	38.35	32.52	32.70
	35.53	34.95	38.70	38.65	33.03	33.24
	35.59	**35.82**	39.19	**39.22**	**33.67**	33.65
20	33.07	33.21	36.21	37.25	31.07	31.33
	34.14	33.33	37.97	37.79	31.73	31.72
	34.02	**34.26**	38.32	**38.40**	32.23	**33.26**
50	27.84	28.37	30.60	33.41	26.56	27.06
	28.33	28.32	34.55	34.28	26.90	27.05
	29.34	**29.72**	35.15	**35.28**	27.99	**28.12**

Table 1. Comparison of video denoising PSNR values (in dB) for several methods. **Top Left**: Patch-based 3D DCT denoising; **Top Right**: sparse K-SVD [19]; **Middle Left**: VBM3D [5]; **Middle Right**: VBM4D [6]; **Bottom Left**: VIDOLSAT with $n = 512$; **Bottom Right**: VIDOLSAT with $n = 768$. For each video and noise level, the best denoising PSNR is marked in bold.

For our VIDOLSAT algorithms, we work with $8 \times 8 \times 8$ ($n = 512$) and $8 \times 8 \times 12$ ($n = 768$) overlapping 3D patches, with $m = 8$ ($m = n_3$) and $m = 12$ respectively. We set spatial overlap stride $v = 1$ for the 3D patches, $\lambda_0 = 1.0 \times 10^{-2}$, $M = 15 \times n$, and $\alpha_{j,i} = 1.9\sigma$ in our experiments. Other parameters such as ρ, K (number of passes), and the estimated noise levels in each pass of the multi-pass scheme were tuned empirically [23, 26]. For (fixed) 3D DCT based denoising, the setting $\alpha_{j,i} = 2.45\sigma$ is used, which was found to work well in our experiments.

To evaluate the performance of the various schemes, we measure the denoised peak signal-to-noise ratio (PSNR) computed between the noiseless reference and the denoised video. Table 1 lists the denoised PSNRs obtained by 3D DCT based denoising, sparse K-SVD denoising, VBM3D, VBM4D, and VIDOLSAT with two different temporal patch sizes. The VIDOLSAT algorithm with $n = 512$ provides average PSNR improvements in Table 1 of 1.35 dB, 0.89 dB, 0.66 dB, and 0.63 dB respectively over the 3D DCT, sparse KSVD, VBM3D, and VBM4D denoising methods respectively. The corresponding improvements provided by VIDOLSAT with $n = 768$ are 1.45 dB, 0.99 dB, 0.76 dB, and 0.72 dB, respectively. With either patch size, VIDOLSAT provides better PSNRs than all of the competing methods for almost all videos and noise levels. Thus our proposed method demonstrates promising performance in video denoising compared to popular competing methods. Moreover, VBM3D and VBM4D are not capable of streaming operation, thus introduce additional latency compared to the proposed online methods.

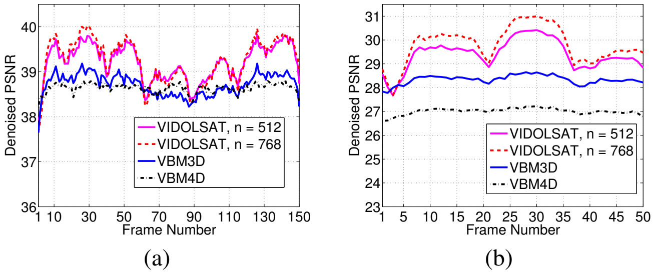

Fig. 2. Frame-by-frame PSNR(dB) of the video (a) *Miss America* with $\sigma = 15$, and (b) *Salesman* with $\sigma = 50$, denoised by the proposed scheme VIDOLSAT ($n = 512$ and $n = 768$), VBM3D and VBM4D, respectively.

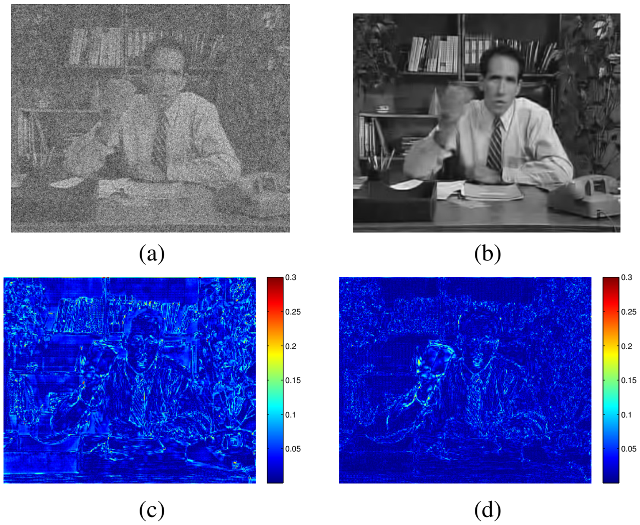

Fig. 3. One frame of *Salesman* denoising result: (a) Noisy frame (PSNR = 14.13 dB), (b) Denoised frame using the proposed VIDOLSAT scheme with $n = 768$ (PSNR = 30.97 dB), (c) Magnitude of error in the denoised frame obtained using VBM4D (PSNR = 27.20 dB), (d) Magnitude of error in (b).

Figure 2 shows the frame-by-frame denoised PSNRs obtained using the VIDOLSAT algorithm for the videos *Miss America* and *Salesman* at $\sigma = 15$ and $\sigma = 50$, respectively, along with the corresponding PSNR values for VBM3D and VBM4D. It is clear that VIDOLSAT outperforms the competing methods for most of the frames. Figure 3 shows one frame of the denoised video *Salesman* at $\sigma = 50$. Comparing 3(c) and 3(d), the denoising result obtained using VIDOLSAT clearly shows lower reconstruction errors than the result obtained from the highly noisy measurements using VBM4D.

5. CONCLUSIONS

In this work, we presented a novel framework for online video denoising. The proposed method uses a temporally sliding window strategy to extract a small set of noisy video frames at each time instant, and then generates sequentially a denoised estimate of these frames with a small and controlled delay (of a few frames), using an efficient online 3D (overlapping) patch-based denoising scheme. Our numerical results demonstrate the promising performance of the proposed method as compared to well-known alternatives such as adaptive overcomplete dictionary-based denoising, VBM3D, VBM4D, or 3D DCT-based denoising.

6. REFERENCES

[1] V. Zlokolica, A. Pizurica, and W. Philips, "Recursive temporal denoising and motion estimation of video," in *Proc. IEEE Int. Conf. Image Proc., ICIP*, 2004, vol. 3, pp. 1465–1468.

[2] F. Jin, P. Fieguth, and L. Winger, "Wavelet video denoising with regularized multiresolution motion estimation," *EURASIP Journal on Advances in Signal Processing*, vol. 2006, pp. 1–11, 2006.

[3] D. Rusanovskyy and K. Egiazarian, "Video denoising algorithm in sliding 3d dct domain," in *Proc. Advanced Concepts for Intelligent Vision Systems*, 2005, pp. 618–625.

[4] N. Rajpoot, Z. Yao, and R. Wilson, "Adaptive wavelet restoration of noisy video sequences," in *Proc. IEEE Int. Conf. Image Proc., ICIP*, 2004, vol. 2, pp. 957–960.

[5] K. Dabov, A. Foi, and K. Egiazarian, "Video denoising by sparse 3d transform-domain collaborative filtering," in *Proc. 15th European Signal Processing Conference*, 2007, pp. 145–149.

[6] M. Maggioni, G. Boracchi, A. Foi, and K. Egiazarian, "Video denoising, deblocking, and enhancement through separable 4-d nonlocal spatiotemporal transforms," *IEEE Trans. Image Process.*, vol. 21, no. 9, pp. 3952–3966, 2012.

[7] M. Elad, P. Milanfar, and R. Rubinstein, "Analysis versus synthesis in signal priors," *Inverse Problems*, vol. 23, no. 3, pp. 947–968, 2007.

[8] A. M. Bruckstein, D. L. Donoho, and M. Elad, "From sparse solutions of systems of equations to sparse modeling of signals and images," *SIAM Review*, vol. 51, no. 1, pp. 34–81, 2009.

[9] R. Rubinstein, T. Faktor, and M. Elad, "K-SVD dictionary-learning for the analysis sparse model," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2012, pp. 5405–5408.

[10] S. Ravishankar and Y. Bresler, "Learning sparsifying transforms," *IEEE Trans. Signal Process.*, vol. 61, no. 5, pp. 1072–1086, 2013.

[11] W. K. Pratt, J. Kane, and H. C. Andrews, "Hadamard transform image coding," *Proc. IEEE*, vol. 57, no. 1, pp. 58–68, 1969.

[12] M. Aharon, M. Elad, and A. Bruckstein, "K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation," *IEEE Trans. Signal Process.*, vol. 54, no. 11, pp. 4311–4322, 2006.

[13] M. Elad and M. Aharon, "Image denoising via sparse and redundant representations over learned dictionaries," *IEEE Trans. Image Process.*, vol. 15, no. 12, pp. 3736–3745, 2006.

[14] K. Skretting and K. Engan, "Recursive least squares dictionary learning algorithm," *IEEE Transactions on Signal Processing*, vol. 58, no. 4, pp. 2121–2130, 2010.

[15] J. Mairal, F. Bach, J. Ponce, G. Sapiro, and A. Zisserman, "Non-local sparse models for image restoration," in *IEEE 12th International Conference on Computer Vision*, 2009, pp. 2272–2279.

[16] B. Ophir, M. Elad, N. Bertin, and M.D. Plumbley, "Sequential minimal eigenvalues - an approach to analysis dictionary learning," in *Proc. European Signal Processing Conference (EUSIPCO)*, 2011, pp. 1465–1469.

[17] M. Yaghoobi, S. Nam, R. Gribonval, and M. E. Davies, "Constrained overcomplete analysis operator learning for cosparse signal modelling," *IEEE Trans. Signal Process.*, vol. 61, no. 9, pp. 2341–2355, 2013.

[18] M. Protter and Michael Elad, "Image sequence denoising via sparse and redundant representations," *IEEE Trans. on Image Processing*, vol. 18, no. 1, pp. 27–36, 2009.

[19] R. Rubinstein, M. Zibulevsky, and M. Elad, "Double sparsity: Learning sparse dictionaries for sparse signal approximation," *IEEE Transactions on Signal Processing*, vol. 58, no. 3, pp. 1553–1564, 2010.

[20] B. K. Natarajan, "Sparse approximate solutions to linear systems," *SIAM J. Comput.*, vol. 24, no. 2, pp. 227–234, Apr. 1995.

[21] G. Davis, S. Mallat, and M. Avellaneda, "Adaptive greedy approximations," *Journal of Constructive Approximation*, vol. 13, no. 1, pp. 57–98, 1997.

[22] S. Ravishankar and Y. Bresler, "Learning doubly sparse transforms for images," *IEEE Trans. Image Process.*, vol. 22, no. 12, pp. 4598–4612, 2013.

[23] S. Ravishankar, B. Wen, and Y. Bresler, "Online sparsifying transform learning - part i: Algorithms," *IEEE Journal of Selected Topics in Signal Process.*, vol. 9, no. 4, pp. 625–636, 2015.

[24] S. Ravishankar and Y. Bresler, "Online sparsifying transform learning - part ii: Convergence analysis," *IEEE Journal of Selected Topics in Signal Process.*, vol. 9, no. 4, pp. 637–646, 2015.

[25] S. Ravishankar, B. Wen, and Y. Bresler, "Online sparsifying transform learning for signal processing," *IEEE Global Conference on Signal and Information Processing*, pp. 364–368, 2014.

[26] B. Wen, S. Ravishankar, and Y. Bresler, "Structured overcomplete sparsifying transform learning with convergence guarantees and applications," *Int. J. Computer Vision*, pp. 1–31, 2014.

[27] D. L. Donoho and I. M. Johnstone, "Ideal spatial adaptation by wavelet shrinkage," *Biometrika*, vol. 81, no. 3, pp. 425–455, 1994.

[28] M. Maggioni, E. Sanchez-Monge, A. Foi, A. Danielyan, K. Dabov, V. Katkovnik, and K. Egiazarian, "BM3D website," `http://www.cs.tut.fi/~foi/GCF-BM3D/`, accessed Dec 2014.

[29] Ron Rubinstein, "Ron Rubinstein web page," `http://www.cs.technion.ac.il/~ronrubin/Software/ksvdsbox11.zip`, accessed Dec 2014.