This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIP.2018.2865684, IEEE Transactions on Image Processing

VIDOSAT: High-dimensional Sparsifying Transform Learning for Online Video Denoising

Bihan Wen, *Student Member, IEEE,* Saiprasad Ravishankar, *Member, IEEE,* and Yoram Bresler, *Fellow, IEEE*

***Abstract*—Techniques exploiting the sparsity of images in a transform domain are effective for various applications in image and video processing. In particular, transform learning methods involve cheap computations and have been demonstrated to perform well in applications such as image denoising and medical image reconstruction. Recently, we proposed methods for online learning of sparsifying transforms from streaming signals, which enjoy good convergence guarantees, and involve lower computational costs than online synthesis dictionary learning. In this work, we apply online transform learning to video denoising. We present a novel framework for online video denoising based on high-dimensional sparsifying transform learning for spatio-temporal patches. The patches are constructed either from corresponding 2D patches in successive frames or using an online block matching technique. The proposed online video denoising requires little memory, and offers efficient processing. Numerical experiments evaluate the performance of the proposed video denoising algorithms on multiple video data sets. The proposed methods outperform several related and recent techniques including denoising with 3D DCT, prior schemes based on dictionary learning, non-local means, background separation, and deep learning, as well as the popular VBM3D and VBM4D.**

***Index Terms*—Sparse representations, Sparsifying transforms, Machine learning, Data-driven techniques, Online learning, Big data, Video denoising.**

I. Introduction

Recent techniques in image and video processing make use of sophisticated models of signals and images. Various properties such as sparsity, low-rankness, etc., have been exploited in inverse problems such as video denoising [1, 2], or other dynamic image reconstruction problems such as magnetic resonance imaging or positron emission tomography [3]. Comparing to fixed signal models, data-driven models and approaches are gaining increasing interest, and lead to promising results in various inverse problems. While the adaptation of synthesis dictionaries for the purpose of denoising image sequences or volumetric data [4, 5] has been studied in some recent papers, the usefulness of learned sparsifying transforms [6, 7] in these applications has not been explored. Video data typically contain correlation along the temporal dimension, which will not be captured by learning sparsifying transforms for the 2D patches of the video frames [2].

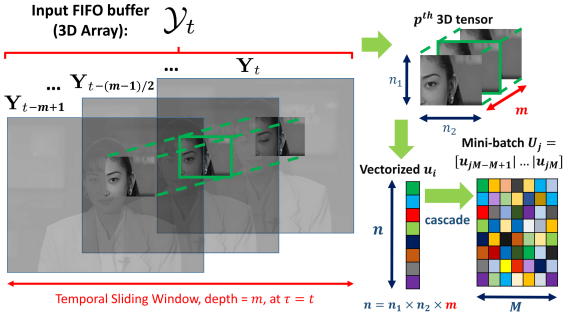


Fig. 1. Video streaming, tensor construction and vectorization.

In this work, we propose a video denoising scheme using high-dimensional online transform learning. We refer to our proposed framework as VIdeo Denoising by Online SpArsifying Transform learning (VIDOSAT). Spatio-temporal (3D) patches are constructed using local 2D patches of the corrupted video, and the sparsifying transform is adapted to denoise these 3D patches on-the-fly. Fig.1 illustrates one way of constructing the (vectorized) spatio-temporal patches or tensors from the streaming video, and Fig.2 is a flow-chart of the proposed VIDOSAT framework. Though we consider 3D spatio-temporal tensors formed by 2D patches for gray-scale video denoising in this work, the proposed denoising methods are also readily extended to higer-dimensional data (e.g., color video [8], hyperspectral images, dynamic MRI).

A. Methodologies and Contributions

While our recent work proposed online transform learning [7], in this work we focus on the application of efficient online sparsifying transform learning to denoising high-dimensional data, particularly videos. We propose video denoising algorithms, named VIDOSAT and VIDOSAT-BM that exploit the underlying data structure and strong spatio-temporal correlations in dynamically varying data. Since these are online and data-adaptive algorithms, the models in VIDOSAT and VIDOSAT-BM change over time and space capturing the changes in data structure and redundancy or correlations. To the best of our knowledge, the proposed methodologies are the first online video denoising methods using adaptive sparse signal modeling, and the first application of high-dimensional sparsifying transform learning to spatio-temporal data. Our methodology and results are summarized as follows:

- The proposed video denoising framework processes noisy frames in an online, sequential fashion to produce stream-

This work was supported in part by the National Science Foundation (NSF) under grant CCF-1320953. Saiprasad Ravishankar was supported in part by the following grants: ONR grant N00014-15-1-2141, DARPA Young Faculty Award D14AP00086, ARO MURI grants W911NF-11-1-0391 and 2015-05174-05, and a UM-SJTU seed grant.

B. Wen, and Y. Bresler are with the Department of Electrical and Computer Engineering and the Coordinated Science Laboratory, University of Illinois, Urbana-Champaign, IL, 61801 USA e-mail: (bwen3, ybresler)@illinois.edu.

S. Ravishankar is with the Department of Electrical Engineering and Computer Science, University of Michigan, Ann Arbor, MI 48109, USA e-mail: ravisha@umich.edu.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIP.2018.2865684, IEEE Transactions on Image Processing

ing denoised video frames. The algorithms require limited storage of a few video frames, and modest computation, scaling linearly with the number of pixels per frame. As such, our methods may be able to handle high definition or high rate video enabling real-time output with controlled delay, using modest computational resources.

- The online transform learning technique exploits the spatio-temporal structure of the video tensors (patches) using adaptive 3D transform-domain sparsity to process them sequentially. The denoised tensors are aggregated to reconstruct the streaming video frames.
- We evaluate the video denoising performance of the proposed algorithms on several datasets, and demonstrate their promising performance compared to several prior or related methods.

A short version of this work appeared in [2]. This work substantially expands upon [2] and proposes novel VIDOSAT methodologies such as involving block matching (BM) to capture motion and rotational redundancy (referred to as VIDOSAT-BM). Moreover, we provide detailed experimental results illustrating the properties of the proposed methods and their performance for several video datasets, with extensive evaluation and comparison to prior or related methods. We also demonstrate the advantages of the proposed VIDOSAT-BM in various scenarios.

Compared to various existing video denoising algorithms, the algorithms in this work (VIDOSAT and VIDOSAT-BM) generate denoised video estimates with improved quality. The key factors to the success of our proposed algorithms are summarized as follows:

- Unlike conventional methods that do not involve learning, the proposed data-driven approaches exploit adaptively estimated video models.
- Transform learning shows performance and speed benefits in video restoration compared to dictionary learning.
- The proposed VIDOSAT algorithms are zero-shot methods, which learn from and restore the noisy data directly.
- The online denoising algorithms with forgetting factors and block matching are particularly effective for handling video with various kinds of motion or dynamics in the scene.
- Learning and restoration of high-dimensional space-time patches fully exploit the spatio-temporal data correlation in video denoising.

B. Major Notation

Vectors (resp. matrices) are denoted by boldface lowercase (resp. uppercase) letters such as $\mathbf{u}$ (resp. $\mathbf{U}$). We use calligraphic uppercase letters (e.g., $\mathcal{U}$) to denote tensors. We denote the vectorization operator for 3D tensors (i.e., for reshaping a 3D array into a vector) as $\mathrm{vec}(\cdot) : \mathbb{R}^{n_1 \times n_2 \times m} \to \mathbb{R}^n$. The vectorized tensor is $\mathbf{u} = \mathrm{vec}(\mathcal{U}) \in \mathbb{R}^n$, with $n = n_1 n_2 m$. Correspondingly, the inverse of the vectorization operator $\mathrm{vec}^{-1}(\cdot) : \mathbb{R}^n \to \mathbb{R}^{n_1 \times n_2 \times m}$ denotes a tensorization operator. The relationship is summarized as follows:

$$\mathcal{U} \in \mathbb{R}^{n_1 \times n_2 \times m} \underset{\mathrm{vec}^{-1}}{\overset{\mathrm{vec}}{\rightleftharpoons}} \mathbf{u} \in \mathbb{R}^n.$$

Indices	**Definition**	**Range**
τ	time index	$1, 2, 3, etc.$
p	spatial index of 3D patches in $\mathcal{Y}_\tau$	$1...P$
i	index of patches within mini-batch	$1...M$
k	local mini-batches index at time τ	$1...N$
j or L_k^τ	global mini-batch index	$1, 2, 3, etc.$
Variables	**Definition**	**Dimension**
$\mathbf{W}_\tau$	adaptive sparsifying transform	$n \times n$
$\mathbf{Y}_\tau$	video frames	$a \times b$
$\mathcal{Y}_\tau$	input FIFO buffer	$a \times b \times m$
$\bar{\mathcal{Y}}_\tau$	output FIFO buffer	$a \times b \times m$
$\mathbf{U}_j$	mini-batch of vectorized data	$n \times M$
$\mathbf{X}_j$	sparse codes of the mini-batch	$n \times M$
$\mathbf{v}_p$	vectorized 3D patch	$n = n_1 n_2 m$
Operators	**Definition**	**Mapping**
R_p	extracts 3D patch in **A1**	$\mathbb{R}^P \to$
B_p	forms 3D patch in **A2** by BM	$\mathbb{R}^{n_1 \times n_2 \times m}$
R_p^*	patch deposit operator in **A1**	$\mathbb{R}^{n_1 \times n_2 \times m}$
B_p^*	patch deposit operator in **A2**	$\to \mathbb{R}^P$

TABLE I
NOTATION OF THE INDICES AND THE MAIN VARIABLES AND OPERATORS. ALGORITHMS **A1** AND **A2** ARE PRESENTED IN SECTION V.

The other major notations of the indices and variables that are used in this work are summarized in Table I. We denote the underlying signal or variable as $\tilde{\mathbf{u}}$, and its noisy measurement (resp. estimate) is denoted as $\mathbf{u}$ (resp. $\hat{\mathbf{u}}$). The other notations used in our algorithms are discussed in later sections.

C. Organization

The rest of the paper is organized as follows. Section II summarizes the related works on video denoising and sparsifying transform learning. Section III briefly discusses the recently proposed formulations for time-sequential signal denoising based on online and mini-batch sparsifying transform learning [7]. Then, Section IV presents the proposed online video processing framework. Section V describes efficient algorithms for the proposed formulations, and specifically, two online approaches for denoising dynamic data. Section VI demonstrates the behavior and promise of the proposed algorithms for denoising several datasets. Section VII concludes with proposals for future work.

II. RELATED WORKS

The proposed VIDOSAT is an online data-driven approach based on learning sparsifying transforms for video denoising. In the following, we briefly review the related works on video denoising and sparsifying transform learning.

A. Video Denoising

Denoising is one of the most important problems in video processing. The ubiquitous use of relatively low-quality smart phone cameras has also led to the increasing importance of

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIP.2018.2865684, IEEE Transactions on Image Processing

Methods	Signal Model			BM	Temp. Corr.	F.G. Sep.
	Fixed	Learned	Online			
fBM3D [11]	✓			✓		
fDnCNN [12]		✓				
3D DCT	✓				✓	
sKSVD		✓			✓	
ReLD [21]		✓			✓	✓
RNLF [18]		✓			✓	
VBM3D [15]	✓			✓	✓	
VBM4D [19]	✓			✓	✓	
VIDOSAT		✓	✓		✓	
VIDOSAT -BM		✓	✓	✓	✓	

TABLE II
COMPARISON BETWEEN VIDEO DENOISING METHODS. FBM3D AND FDNCNN APPLY BM3D AND DNCNN, RESPECTIVELY, TO DENOISE INDIVIDUAL FRAMES. VIDOSAT AND VIDOSAT-BM ARE THE METHODS PROPOSED HERE, WHILE THE 3D DCT METHOD APPLIES THE VIDOSAT FRAMEWORK BUT USES THE FIXED 3D DCT TRANSFORM. WE ABBREVIATE TEMPORAL CORRELATION (TEMP. CORR.) AND FOREGROUND SEPARATION (F.G. SEP.,).

video denoising. Recovering high-quality video also improves robustness in high-level vision tasks [9, 10].

Though image denoising algorithms, such as the popular BM3D [11] or the recent DnCNN [12] methods can be applied to each video frame independently, most of the video denoising techniques (or more generally, methods for reconstructing dynamic data from measurements [13]) exploit the spatio-temporal correlation in dynamic image sequences. Natural videos have local structures that are sparse or compressible in some transform domain, or in certain dictionaries, e.g., discrete cosine transform (DCT) [14] and wavelets [1]. Prior works exploited this fact and proposed video (or high-dimensional data) denoising algorithms based on adaptive sparse approximation [5] and Wiener filtering [15]. Besides, recent works also applied Bayesian modeling [16], low-rankness [17], and non-local means [18] for video denoising.

Different from images or volumetric data, videos typically involve various kinds of motion or dynamics in the scene, e.g., moving objects or people, rotations, etc. State-of-the-art video and image denoising algorithms utilize block matching (BM) to group local patches over space and time (to account for motion), and apply denoising jointly for such matched data [11, 15, 19]. Alternatively, recent works [20, 21] proposed to separate the sparse and dynamic foreground from the low-rank background in video denoising, modeling their properties differently. Table II summarizes the key attributes of the popular and related video denoising methods as well as the proposed methods. The methods classified as using a fixed signal model use an analytical sparsifying transform. Compared to the prior approaches, our techniques exploit several properties such as spatio-temporal data redundancy, block matching, online learning, and forgetting factors to denoise videos effectively.

B. Sparsifying Transform Learning

Many of the aforementioned video denoising methods exploit sparsity in a fixed transform domain (e.g., DCT) as part of their framework. Several recent works have shown that the data-driven adaptation of sparse signal models (e.g., based on training signals, or directly from corrupted measurements) usually leads to high quality results (e.g., compared to fixed or analytical models) in many applications [4, 22–30]. Synthesis dictionary learning is the best-known adaptive sparse representation technique [22, 31]. However, obtaining optimal sparse representations of signals in synthesis dictionary models, known as synthesis sparse coding, is NP-hard (Non-deterministic Polynomial-time hard) in general. The commonly used approximate sparse coding algorithms [32–35] typically still involve relatively expensive computations for large-scale problems.

As an alternative, the sparsifying transform model suggests that the signal $\mathbf{u}$ is approximately sparsifiable using a transform $\mathbf{W} \in \mathbb{R}^{m \times n}$, i.e., $\mathbf{W}\mathbf{u} = \mathbf{x} + \mathbf{e}$, with $\mathbf{x} \in \mathbb{R}^m$ a vector with few non-zeros, called the transform sparse code and $\mathbf{e}$ a modeling error term in the transform domain. A key advantage of this model over the synthesis dictionary model, is that for a given transform $\mathbf{W}$, the optimal sparse approximation $\mathbf{x}$ of sparsity level s minimizing the modeling error $\|\mathbf{e}\|_2$ is obtained *exactly and cheaply* by simple thresholding of $\mathbf{W}\mathbf{u}$ to its s largest magnitude components. Another advantage is that with $\mathbf{u}$ being given data, the transform model does not involve a product between $\mathbf{W}$ and unknown variables, so learning algorithms for $\mathbf{W}$ can be simpler and more reliable. Recent works ([6] and references therein) proposed learning sparsifying transforms with cheap algorithms that alternate between updating the sparse approximations of training signals in a transform domain using simple thresholding-based transform sparse coding, and efficiently updating the sparsifying transform. Transform learning-based techniques have been shown to be useful in various applications such as sparse data representations, image denoising, inpainting, segmentation, magnetic resonance imaging (MRI), and computed tomography (CT) [27, 36, 37].

In prior works on batch transform learning [6, 27, 36], the transform was adapted using all the training data, which is efficient and comes with convergence guarantees. When processing large-scale streaming data, it is also important to compute results online, or sequentially over time. Our recent work [7] proposed online transform learning, which sequentially adapts the sparsifying transform and transform-sparse coefficients for sequentially processed signals. This approach involves cheap computation and limited memory requirements. Compared to popular techniques for online synthesis dictionary learning [38], the online adaptation of sparsifying transforms allows for cheaper and exact updates [7], and is thus well suited for high-dimensional data applications.

III. SIGNAL DENOISING VIA ONLINE TRANSFORM LEARNING

The goal in denoising is to recover an estimate of a signal $\tilde{\mathbf{u}} \in \mathbb{R}^n$ from the measurement $\mathbf{u} = \tilde{\mathbf{u}} + \mathbf{e}$, corrupted by additive noise $\mathbf{e}$. Here, we consider a time sequence of noisy measurements $\{\mathbf{u}_t\}$, with $\mathbf{u}_t = \tilde{\mathbf{u}}_t + \mathbf{e}_t$. We assume noise $\mathbf{e}_t \in \mathbb{R}^n$ whose entries are independent and identically distributed (i.i.d.) Gaussian with zero mean and possibly

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIP.2018.2865684, IEEE Transactions on Image Processing

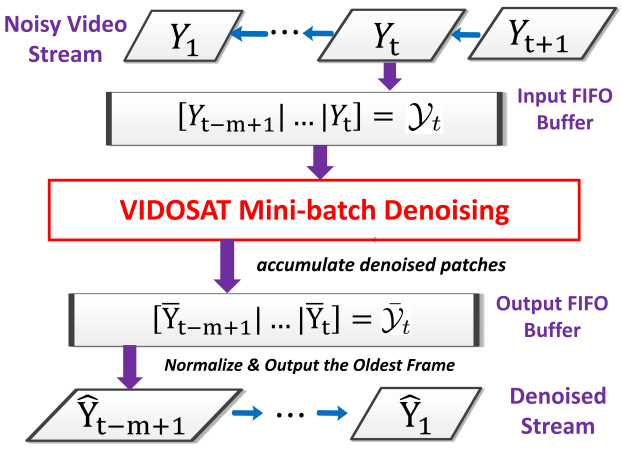


Fig. 2. Illustration of the online video streaming and denoising framework.

time-varying but known variance σ_t^2. Online denoising is to recover the estimates $\hat{\mathbf{u}}_t$ for $\tilde{\mathbf{u}}_t \ \forall \ t$ sequentially. Such time-sequential denoising with low memory requirements would be especially useful for streaming data applications. We assume that the underlying signals $\{\tilde{\mathbf{u}}_t\}$ are approximately sparse in an (unknown, or to be estimated) transform domain.

A. Online Transform Learning

In prior work [7], we proposed an online signal denoising methodology based on sparsifying transform learning, where the transform is adapted based on sequentially processed data. For time $t = 1, 2, 3$, etc, the problem of updating the adaptive sparsifying transform and sparse code (i.e., the sparse representation in the adaptive transform domain) to account for the new noisy signal $\mathbf{u}_t \in \mathbb{R}^n$ is

$$\left\{\hat{\mathbf{W}}_t, \hat{\mathbf{x}}_t\right\} = \underset{\mathbf{W},\, \mathbf{x}_t}{\arg\min} \frac{1}{t} \sum_{\tau=1}^{t} \left\{ \|\mathbf{W}\mathbf{u}_\tau - \mathbf{x}_\tau\|_2^2 + \lambda_\tau \nu(\mathbf{W}) \right\} + \frac{1}{t} \sum_{\tau=1}^{t} \alpha_\tau^2 \|\mathbf{x}_\tau\|_0 \quad s.t. \ \mathbf{x}_\tau = \hat{\mathbf{x}}_\tau, \ 1 \leq \tau \leq t-1 \quad \text{(P1)}$$

where the ℓ_0 "norm" counts the number of nonzeros in x_τ, which is the sparse code of $\mathbf{u}_\tau$. Thus $\|\mathbf{W}\mathbf{u}_\tau - \mathbf{x}_\tau\|_2^2$ is the sparsification error (i.e., the modeling error in the transform model) for $\mathbf{u}_\tau$ in the transform $\mathbf{W} \in \mathbb{R}^{n \times n}$. The term $\nu(\mathbf{W}) = -\log|\det \mathbf{W}| + \|\mathbf{W}\|_F^2$ is a transform learning regularizer [7], $\lambda_\tau = \lambda_0 \|\mathbf{u}_\tau\|_2^2$ with $\lambda_0 > 0$ allows the regularizer term to scale with the first term in the cost, and the weight α_τ is chosen proportional to σ_τ (the standard deviation of noise in $\mathbf{u}_\tau$). Matrix $\hat{\mathbf{W}}_t$ in (P1) is the optimal transform at time t, and $\hat{\mathbf{x}}_t$ is the optimal sparse code for $\mathbf{u}_t$.

Note that at time t, only the latest optimal sparse code $\hat{\mathbf{x}}_t$ is updated in (P1)[1] along with the transform $\hat{\mathbf{W}}_t$. The condition $\mathbf{x}_\tau = \hat{\mathbf{x}}_\tau, \ 1 \leq \tau \leq t-1$, is therefore assumed. For brevity, we will not explicitly restate this condition (or, its variants) in the formulations in the rest of this paper. Although at each time t the transform is updated based on all the past and present observed data, the online algorithm for (P1) [7] involves efficient operations based on a few matrices of modest size, accumulated sequentially over time.

The regularizer $\nu(\mathbf{W})$ in (P1) prevents trivial solutions and controls the condition number and scaling of the learnt transform [7]. The condition number $\kappa(\mathbf{W})$ is upper bounded by a monotonically increasing function of $\nu(\mathbf{W})$ [7]. In the limit $\lambda_0 \to \infty$ (and assuming the $\mathbf{u}_\tau$, $1 \leq \tau \leq t$, are not all zero), the condition number of the optimal transform in (P1) tends to 1. The specific choice of λ_0 (and hence the condition number) depends on the application.

1) Denoising: Given the optimal transform $\hat{\mathbf{W}}_t$ and the sparse code $\hat{\mathbf{x}}_t$, a simple estimate of the denoised signal is obtained as $\hat{\mathbf{u}}_t = \hat{\mathbf{W}}_t^{-1}\hat{\mathbf{x}}_t$. Online transform learning can also be used for patch-based denoising of large images [7]. Overlapping patches of the noisy images are processed sequentially (e.g., in raster scan order) via (P1), and the denoised image is obtained by averaging together the denoised patches at their respective image locations.

2) Forgetting factor: For non-stationary or highly dynamic data, it may not be desirable to uniformly fit a single transform $\mathbf{W}$ to all the $\mathbf{u}_\tau$, $1 \leq \tau \leq t$, in (P1). Such data can be handled by introducing a forgetting factor $\rho^{t-\tau}$ (with a constant $0 < \rho < 1$) that scales the terms in (P1) [7]. The forgetting factor diminishes the influence of "old" data. The objective function in this case is modified as

$$\frac{1}{C_t} \sum_{\tau=1}^{t} \rho^{t-\tau} \left\{ \|\mathbf{W}\mathbf{u}_\tau - \mathbf{x}_\tau\|_2^2 + \lambda_\tau \nu(\mathbf{W}) + \alpha_\tau^2 \|\mathbf{x}_\tau\|_0 \right\}. \quad (1)$$

where $C_t = \sum_{\tau=1}^{t} \rho^{t-\tau}$ is the normalization factor.

B. Mini-batch learning

Another useful variation of Problem (P1) involves *mini-batch* learning, where a block (group), or *mini-batch* of signals is processed at a time [7]. Assuming a fixed mini-batch size M, the Lth ($L \geq 1$) mini-batch of signals is $\mathbf{U}_L = \left[\mathbf{u}_{LM-M+1} \,|\, \mathbf{u}_{LM-M+2} \,|\, \dots \,|\, \mathbf{u}_{LM}\right]$. For $L = 1, 2, 3$, etc, the mini-batch sparsifying transform learning problem is

$$\text{(P2)} \quad \left\{\hat{\mathbf{W}}_L, \hat{\mathbf{X}}_L\right\} = \underset{\mathbf{W},\, \mathbf{X}_L}{\arg\min} \frac{1}{LM} \sum_{j=1}^{L} \|\mathbf{W}\mathbf{U}_j - \mathbf{X}_j\|_F^2 + \frac{1}{LM} \sum_{l=1}^{LM} \alpha_l^2 \|\mathbf{x}_l\|_0 + \frac{1}{LM} \sum_{j=1}^{L} \Lambda_j \, \nu(\mathbf{W})$$

where the regularizer weight is $\Lambda_j = \lambda_0 \left\|\mathbf{U}_j\right\|_F^2$, and the matrix $\mathbf{X}_L = \left[\mathbf{x}_{LM-M+1} \,|\, \mathbf{x}_{LM-M+2} \,|\, \dots \,|\, \mathbf{x}_{LM}\right]$ contains the block of sparse codes corresponding to $\mathbf{U}_L$.

Since we only consider a finite number of frames or patches in practice (e.g., in the proposed VIDOSAT algorithms), the normalizations by $1/t$ in (P1), $1/C_t$ in (1), and $1/LM$ in (P2) correspondingly have no effect on the optimum $\left\{\hat{\mathbf{W}}_t, \hat{\mathbf{X}}_t\right\}$ or $\left\{\hat{\mathbf{W}}_L, \hat{\mathbf{X}}_L\right\}$. Thus we drop, for clarity[2], normalization factors from (P3) and all subsequent expressions for the cost functions.

[1]This is because only the signal $\mathbf{u}_t$ is assumed to be stored in memory at time t for the online scheme.

[2]In practice, such normalizations may still be useful, to control the dynamic range of various internal variables in the algorithm.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIP.2018.2865684, IEEE Transactions on Image Processing

Once (P2) is solved, a simple denoised estimate of the noisy block of signals in $\mathbf{U}_L$ is obtained as $\hat{\mathbf{U}}_L = \hat{\mathbf{W}}_L^{-1}\hat{\mathbf{X}}_L$. The mini-batch transform learning Problem (P2) is a generalized version of (P1), with (P2) being equivalent to (P1) for $M = 1$. Similar to (1), (P2) can be modified to include a forgetting factor. 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 (i.e., delay in producing output) [7].

IV. VIDOSAT Framework and Formulations

Prior work on adaptive sparsifying transform-based image denoising [6, 7, 27] adapted the transform operator to 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 [5]. We therefore propose an online approach to video denoising by learning a sparsifying transform on appropriately constructed 3D spatio-temporal patches.

A. Video Streaming and Denoising Framework

Fig. 2 illustrates the framework of our proposed online denoising scheme for streaming videos. The frames $\mathbf{Y}_\tau \in R^{a\times b}$ of the noisy video (assumed to be corrupted by additive i.i.d. Gaussian noise) arrive at $\tau = 1, 2, \ldots$. At time $\tau = t$, the newly arrived frame $\mathbf{Y}_t$ is added to a fixed-size FIFO (first in first out) buffer (i.e., queue) that stores a block of m consecutive frames $\{\mathbf{Y}_i\}_{i=t-m+1}^{t}$. The oldest (leftmost) frame is dropped from the buffer at each time instant. We denote the spatio-temporal tensor or 3D array obtained by stacking noisy frames along the temporal dimension in the buffer as $\mathcal{Y}_t = [\mathbf{Y}_{t-m+1} \mid \ldots \mid \mathbf{Y}_t] \in \mathbb{R}^{a\times b\times m}$. We denoise the noisy array $\mathcal{Y}_t$ using the proposed VIDOSAT mini-batch denoising algorithms (denoted by the red box in Fig. 2) that are discussed in Sections IV-B and V. These algorithms denoise groups (mini-batches) of 3D patches sequentially and adaptively, by learning sparsifying transforms. Overlapping patches are used in our framework.

The patches output by the mini-batch denoising algorithms are deposited at their corresponding spatio-temporal locations in the fixed-size FIFO output buffer $\bar{\mathcal{Y}}_t = [\bar{\mathbf{Y}}_{t-m+1} \mid \ldots \mid \bar{\mathbf{Y}}_t]$ by adding them to the contents of $\bar{\mathcal{Y}}_t$. We call this process *patch aggregation*. The streaming scheme then outputs the oldest frame $\bar{\mathbf{Y}}_{t-m+1}$. The denoised estimate $\hat{\mathbf{Y}}_{t-m+1}$ is obtained by normalizing $\bar{\mathbf{Y}}_{t-m+1}$ pixel-wise by the number of occurrences of each pixel in the aggregated patches. (see Section V for details).

Though any frame could be denoised and output from $\bar{\mathcal{Y}}_t$ instantaneously, we observe improved denoising quality by averaging over multiple denoised estimates obtained at different times. Fig. 3 illustrates how the output buffer varies from time t to $t + (m - 1)$, to output the denoised $\hat{\mathbf{Y}}_t$. In practice, we set the length of the output buffer $\bar{\mathcal{Y}}$ to be the same as the 3D patch depth m, such that each denoised frame $\hat{\mathbf{Y}}_t$ is output by averaging over its estimates from all 3D patches that group the tth frame with $m-1$ adjacent frames. We refer

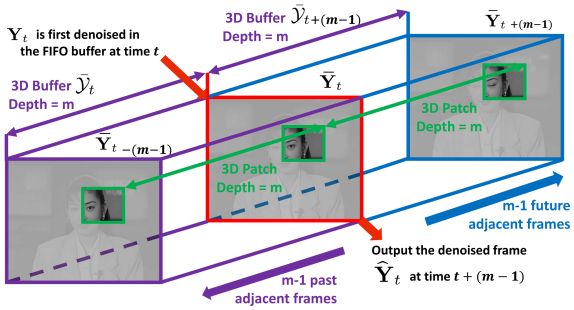


Fig. 3. Evolution of the output buffer from time t to $t + (m - 1)$ used to generate the denoised frame output $\hat{\mathbf{Y}}_t$.

to this scheme as "two-sided" denoising, since the tth frame is denoised together with both past and future adjacent frames ($m - 1$ frames on each side), which are highly correlated. Now, data from frame $\mathbf{Y}_t$ is contained in 3D patches that also contain data from frame $\mathbf{Y}_{t+m-1}$. Once these patches are denoised, they will contribute (by aggregation into the output buffer) to the final denoised frame $\hat{\mathbf{Y}}_t$. Therefore, we must wait for frame $\mathbf{Y}_{t+m-1}$ before producing the final estimate $\hat{\mathbf{Y}}_t$. Thus there is a delay of $m-1$ frames between the arrival of the noisy $\mathbf{Y}_t$ and the generation of its final denoised estimate $\hat{\mathbf{Y}}_t$.

B. VIDOSAT Mini-Batch Denoising Formulation

Here, we discuss the mini-batch denoising formulation that is a core part of the proposed online video denoising framework. For each time instant t, we denoise P partially overlapping size $n_1 \times n_2 \times m$ 3D patches of $\mathcal{Y}_t$ whose vectorized versions are denoted as $\{\mathbf{v}_p^t\}_{p=1}^{P}$, with $\mathbf{v}_p^t \in \mathbb{R}^n$, $n = mn_1n_2$. We sequentially process disjoint groups of M such patches, and the groups or mini-batches of patches (total of N mini-batches, where $P = MN$) are denoted as $\{\mathbf{U}_{L_k^t}\}_{k=1}^{N}$, with $\mathbf{U}_{L_k^t} \in \mathbb{R}^{n\times M}$. Here, k is the *local* mini-batch index within the set of P patches of $\mathcal{Y}_t$, whereas $L_k^t \triangleq N\times(t-1)+k$ is the *global* mini-batch index, identifiying the mini-batch in both time t and location within the set of P patches of $\mathcal{Y}_t$.

For each t, we solve the following online transform learning problem for each $k = 1, 2, 3, ..., N$, to adapt the transform and sparse codes sequentially to the mini-batches in $\mathcal{Y}_t$:

$$\text{(P3)} \quad \left\{\hat{\mathbf{W}}_{L_k^t}, \hat{\mathbf{X}}_{L_k^t}\right\} = \underset{\mathbf{W},\mathbf{X}_{L_k^t}}{\arg\min} \sum_{j=1}^{L_k^t} \rho^{L_k^t-j} \left\|\mathbf{W}\mathbf{U}_j - \mathbf{X}_j\right\|_F^2 + \sum_{j=1}^{L_k^t} \rho^{L_k^t-j} \left\{\Lambda_j \, \nu(\mathbf{W}) + \sum_{i=1}^{M} \alpha_{j,i}^2 \left\|\mathbf{x}_{j,i}\right\|_0\right\}.$$

Here, the transform is adapted based on patches from *all* the *observed* $\mathbf{Y}_\tau$, $1 \le \tau \le t$. The matrix $\mathbf{X}_j = [\mathbf{x}_{j,1} \mid \ldots \mid \mathbf{x}_{j,M}] \in \mathbb{R}^{n\times M}$ denotes the transform sparse codes corresponding to the mini-batch $\mathbf{U}_j$. The sparsity penalty weight $\alpha_{j,i}^2$ in (P3) controls the number of non-zeros in $\mathbf{x}_{j,i}$. We set $\alpha_{j,i} = \alpha_0\sigma_{j,i}$, where $\alpha_0 > 0$ is a constant and $\sigma_{j,i}$ is the noise standard deviation for each patch. We use a forgetting

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIP.2018.2865684, IEEE Transactions on Image Processing

factor $\rho^{L_k^t - j}$ in (P3) to diminish the influence of old frames and old mini-batches.

Once (P3) is solved, the denoised version of the current noisy mini-batch $\hat{\mathbf{U}}_{L_k^t}$ is computed. The columns of the denoised $\hat{\mathbf{U}}_{L_k^t}$ are tensorized and aggregated at the corresponding spatial and temporal locations in the output FIFO buffer. Section V next discusses the proposed VIDOSAT algorithms with two different ways of constructing the 3D patches in full detail.

V. Video Denoising Algorithms

We now discuss two video denoising algorithms, namely VIDOSAT and VIDOSAT-BM. VIDOSAT-BM uses block matching to generate the 3D patches from $\mathcal{Y}_t$. Though these methods differ in the way they construct the 3D patches, and the way the denoised patches are aggregated in the output FIFO, they both denoise groups of 3D patches sequentially by solving (P3). The VIDOSAT denoising algorithm (without BM) is summarized in Algorithm **A1**[3]. The VIDOSAT-BM algorithm, a modified version of Algorithm **A1**, is discussed in Section V-B.

A. VIDOSAT

As discussed in Section IV-B, the VIDOSAT algorithm processes each mini-batch $\mathbf{U}_j$ in $\mathcal{Y}_\tau$ sequentially. We solve the mini-batch transform learning problem (P3) using a simple alternating minimization approach, with one alternation per mini-batch, which works well and saves computation. Initialized with the most recently estimated transform (warm start), we perform two steps for (P3): Sparse Coding, and Mini-batch Transform Update, which compute $\hat{\mathbf{X}}_j$ and update $\hat{\mathbf{W}}_j$, respectively. Then, we compute the denoised mini-batch $\hat{\mathbf{U}}_j$, and aggregate the denoised patches into the output buffer $\bar{\mathcal{Y}}_\tau$.

The major steps of the VIDOSAT algorithm **A1** for denoising the kth mini-batch $\mathbf{U}_{L_k^t}$ at time t and further processing these denoised patches are described. To facilitate the exposition and interpretation in terms of the general online denoising algorithm described, various quantities (such as positions of 3D patches in the video stream) are indexed in the text with respect to absolute time t. On the other hand, to emphasize the streaming nature of Algorithm **A1** and its finite (and modest) memory requirements, indexing of internal variables in the statement of the algorithm is local.

1) Noisy Mini-Batch Formation: To construct each mini-batch $\mathbf{U}_{L_k^t}$, partially overlapping $n_1 \times n_2 \times m$ 3D patches of $\mathcal{Y}_t$ are extracted sequentially in a spatially contiguous order (raster scan order with direction reversal on each line)[4]. Let $R_p\mathcal{Y}_t$ denote the pth vectorized 3D patch of $\mathcal{Y}_t$, with R_p being the patch-extraction operator. Considering the patch indices $S_k = \{M(k-1)+1, ..., Mk\}$ for the kth mini-batch, we extract $\left\{\mathbf{v}_p^t = \mathrm{vec}(R_p\mathcal{Y}_t)\right\}_{p\in S_k}$ as the patches in the mini-batch. Thus $\mathbf{U}_{L_k^t} = \left[\mathbf{v}_{M(k-1)+1}^t \mid ... \mid \mathbf{v}_{Mk}^t\right]$. To impose spatio-temporal contiguity of 3D patches extracted from two adjacent stacks of frames, we reverse the raster scan order (of patches) between $\mathcal{Y}_t$ and $\mathcal{Y}_{t+1}$.

2) Sparse Coding: Given the sparsifying transform $\mathbf{W} = \hat{\mathbf{W}}_{L_k^t-1}$ estimated for the most recent mini-batch, we solve Problem (P3) for the sparse coefficients $\hat{\mathbf{X}}_{L_k^t}$:

$$\hat{\mathbf{X}}_{L_k^t} = \underset{\mathbf{X}}{\arg\min} \left\|\mathbf{W}\mathbf{U}_{L_k^t} - \mathbf{X}\right\|_F^2 + \sum_{i=1}^{M} \alpha_{L_k^t,i}^2 \left\|\mathbf{x}_i\right\|_0 \quad (2)$$

A solution for (2) is given in closed-form as $\hat{\mathbf{x}}_{L_k^t,i} = H_{\alpha_{L_k^t,i}}(\hat{\mathbf{W}}_{L_k^t-1}\mathbf{u}_{L_k^t,i})\ \forall\ i$ [7]. Here, the hard thresholding operator $H_\alpha(\cdot) : \mathbb{R}^n \to \mathbb{R}^n$ is applied to a vector element-wise, as defined by

$$\left(H_\alpha(\mathbf{d})\right)_r = \begin{cases} 0 & ,\ |d_r| < \alpha \\ d_r & ,\ |d_r| \geq \alpha \end{cases} \quad (3)$$

This simple hard thresholding operation for transform sparse coding is similar to traditional techniques involving analytical sparsifying transforms [40].

Algorithm **A1**: VIDOSAT Denoising Algorithm

Input: The noisy frames $\mathbf{Y}_\tau$ ($\tau = 1, 2, 3,$ etc.), and the initial transform $\mathbf{W}_0$ (e.g., 3D DCT).
Initialize: $\hat{\mathbf{W}} = \mathbf{W}_0$, $\mathbf{\Gamma} = \mathbf{\Theta} = \mathbf{0}$, $\beta = 0$,
and output buffer $\bar{\mathcal{Y}} = 0$.
For $\tau = 1, 2, 3,$ **etc., Repeat**
The newly arrived frame $\mathbf{Y}_\tau \to$ latest frame in the input FIFO frame buffer $\mathcal{Y}$.
For $k = 1, ..., N$ **Repeat**
Indices of patches in $\mathcal{Y}$: $S_k = \{M(k-1)+1, ..., Mk\}$.
1) **Noisy Mini-Batch Formation:**
   a) Patch Extraction: $\mathbf{v}_p = \mathrm{vec}(R_p\mathcal{Y})\quad \forall p \in S_k$.
   b) $\mathbf{U} = \left[\mathbf{u}_1 \mid ... \mid \mathbf{u}_M\right] \leftarrow \left[\mathbf{v}_{Mk-M+1} \mid ... \mid \mathbf{v}_{Mk}\right]$.
2) **Sparse Coding:** $\hat{\mathbf{x}}_i = H_{\alpha_i}(\hat{\mathbf{W}}\mathbf{u}_i)\ \ \forall i \in \{1, ..., M\}$.
3) **Mini-batch Transform Update:**
   a) Define $\Lambda \triangleq \lambda_0 \|\mathbf{U}\|_F^2$ and $\hat{\mathbf{X}} \triangleq \left[\hat{\mathbf{x}}_1 \mid ... \mid \hat{\mathbf{x}}_M\right]$.
   b) $\mathbf{\Gamma} \leftarrow \rho\mathbf{\Gamma} + \mathbf{U}\mathbf{U}^T$.
   c) $\mathbf{\Theta} \leftarrow \rho\mathbf{\Theta} + \mathbf{U}\hat{\mathbf{X}}^T$.
   d) $\beta \leftarrow \rho\beta + \Lambda$.
   e) Matrix square root: $\mathbf{Q} \leftarrow (\mathbf{\Gamma} + \beta\mathbf{I})^{1/2}$.
   f) Full SVD: $\mathbf{\Phi}\mathbf{\Sigma}\mathbf{\Psi}^T \leftarrow \mathrm{SVD}(\mathbf{Q}^{-1}\mathbf{\Theta})$.
   g) $\hat{\mathbf{W}} \leftarrow 0.5\mathbf{\Psi}\left(\mathbf{\Sigma} + \left(\mathbf{\Sigma}^2 + 2\beta\mathbf{I}\right)^{\frac{1}{2}}\right)\mathbf{\Phi}^T\mathbf{Q}^{-1}$.
4) **3D Denoised Patch Reconstruction:**
   a) Update Sparse Codes: $\hat{\mathbf{x}}_i = H_{\alpha_i}(\hat{\mathbf{W}}\mathbf{u}_i)\ \ \forall i$.
   b) Denoised mini-batch: $\hat{\mathbf{U}} = \hat{\mathbf{W}}^{-1}\hat{\mathbf{X}}$.
   c) $\left[\hat{\mathbf{v}}_{M(k-1)+1} \mid ... \mid \hat{\mathbf{v}}_{Mk}\right] \leftarrow \hat{\mathbf{U}}$
   d) Tensorization: $\hat{\mathcal{V}}_p = \mathrm{vec}^{-1}(\hat{\mathbf{v}}_p)\ \forall p \in S_k$.
5) **Aggregation:** Aggregate patches $\left\{\hat{\mathcal{V}}_p\right\}$ at corresponding locations: $\bar{\mathcal{Y}} \leftarrow \sum_{p\in S_k} R_p^*\hat{\mathcal{V}}_p$.
End
Output: The oldest frame in $\bar{\mathcal{Y}}$ after normalization $\to$ the denoised frame $\hat{\mathbf{Y}}_{\tau-m+1}$.
End

---

[3]In practice, we wait for the first m frames to be received, before starting Algorithm **A1**, to avoid zero frames in the input FIFO buffer

[4]We did not observe any marked improvement in denoising performance, when using other scan orders such as raster or Peano-Hilbert scan [39].

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIP.2018.2865684, IEEE Transactions on Image Processing

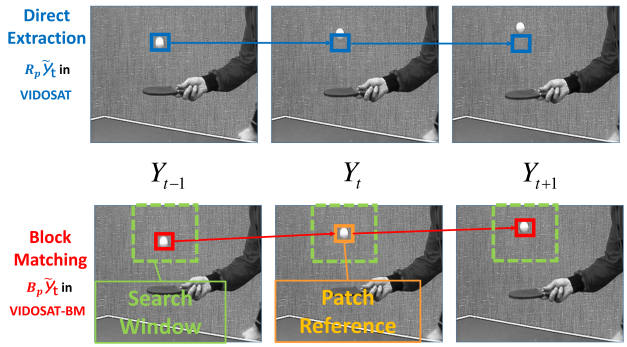


Fig. 4. Different 3D patch construction methods in VIDOSAT (top) and VIDOSAT-BM (bottom). The 3D search window used in VIDOSAT-BM in green broken line.

3) Mini-batch Transform Update: We solve Problem (P3) for $\mathbf{W}$ with fixed $\mathbf{X}_j = \hat{\mathbf{X}}_j$, $1 \leq j \leq L_k^t$, as follows:

$$\min_{\mathbf{W}} \sum_{j=1}^{L_k^t} \rho^{L_k^t - j} \left\{ \left\| \mathbf{W}\mathbf{U}_j - \mathbf{X}_j \right\|_F^2 + \Lambda_j \nu(\mathbf{W}) \right\}. \quad (4)$$

This problem has a simple solution (similar to Section III-B2 in [7]). Set index $J \triangleq L_t^k$, and define the following quantities: $\mathbf{\Gamma}_J \triangleq \sum_{j=1}^{J} \rho^{J-j} \mathbf{U}_j \mathbf{U}_j^T$, $\mathbf{\Theta}_J \triangleq \sum_{j=1}^{J} \rho^{J-j} \mathbf{U}_j \hat{\mathbf{X}}_j^T$, and $\beta_J \triangleq \sum_{j=1}^{J} \rho^{J-j} \Lambda_j$. Let $\mathbf{Q} \in \mathbb{R}^{n \times n}$ be a square root (e.g., Cholesky factor) of $(\mathbf{\Gamma}_J + \beta_J \mathbf{I})$, i.e., $\mathbf{Q}\mathbf{Q}^T = \mathbf{\Gamma}_J + \beta_J \mathbf{I}$. Denoting the full singular value decomposition (SVD) of $\mathbf{Q}^{-1}\mathbf{\Theta}_J$ as $\mathbf{\Phi}\mathbf{\Sigma}\mathbf{\Psi}^T$, based on [7] the closed-form solution to (4) is

$$\hat{\mathbf{W}}_J = 0.5\mathbf{\Psi} \left(\mathbf{\Sigma} + \left(\mathbf{\Sigma}^2 + 2\beta_J \mathbf{I} \right)^{\frac{1}{2}} \right) \mathbf{\Phi}^T \mathbf{Q}^{-1} \quad (5)$$

where $\mathbf{I}$ denotes the identity matrix, and $(\cdot)^{\frac{1}{2}}$ denotes the positive definite square root of a positive definite (diagonal) matrix. The quantities $\mathbf{\Gamma}_J$, $\mathbf{\Theta}_J$, and β_J are all computed sequentially over time t and mini-batches k [7].

4) 3D Denoised Patch Reconstruction: We denoise $\mathbf{U}_{L_k^t}$ using the updated transform. First, we repeat the sparse coding step using the updated $\hat{\mathbf{W}}_{L_k^t}$ as $\hat{\mathbf{x}}_{L_k^t,i} = H_{\alpha_{L_k^t,i}}(\hat{\mathbf{W}}_{L_k^t} \mathbf{u}_{L_k^t,i})$ $\forall$ i. Then, with fixed $\hat{\mathbf{W}}_{L_k^t}$ and $\hat{\mathbf{X}}_{L_k^t}$, the denoised mini-batch is obtained in the least squares sense under the transform model as

$$\hat{\mathbf{U}}_{L_k^t} = \hat{\mathbf{W}}_{L_k^t}^{-1} \hat{\mathbf{X}}_{L_k^t}. \quad (6)$$

The denoised mini-batch is used to update the denoised (vectorized) 3D patches as $\hat{\mathbf{v}}_{M(k-1)+i}^t = \hat{\mathbf{u}}_{L_k^t,i}$ $\forall i$. All reconstructed vectors $\{\hat{\mathbf{v}}_p^t\}_{p \in S_k}$ from the kth mini-batch denoising result are tensorized as $\{\text{vec}^{-1}(\hat{\mathbf{v}}_p^t)\}_{p \in S_k}$.

5) Aggregation: The denoised 3D patches $\{\text{vec}^{-1}(\hat{\mathbf{v}}_p^t)\}_{p \in S_k}$ from each mini-batch are sequentially aggregated at their corresponding spatial and temporal locations in the output FIFO buffer as $\sum_{p \in S_k} R_p^* \,\text{vec}^{-1}(\hat{\mathbf{v}}_p^t) \rightarrow \bar{\mathcal{Y}}_t \in \mathbb{R}^{a \times b \times m}$, where the adjoint R_p^* is the patch deposit operator. Fig. 5 illustrates the patch deposit procedure for aggregation.

When all N denoised mini-batches for $\mathcal{Y}_t$ are generated, and the patch aggregation in $\bar{\mathcal{Y}}_t$ completes, the oldest frame in $\bar{\mathcal{Y}}_t$ is normalized pixel-wise by the number of occurrences (which ranges from $2m-1$, for pixels at the corners of a video frame, to n for pixels away from the borders of a video frame) of that pixel among patches aggregated into the output buffer. This normalized result is output as the denoised frame $\hat{\mathbf{Y}}_{t-m+1}$.

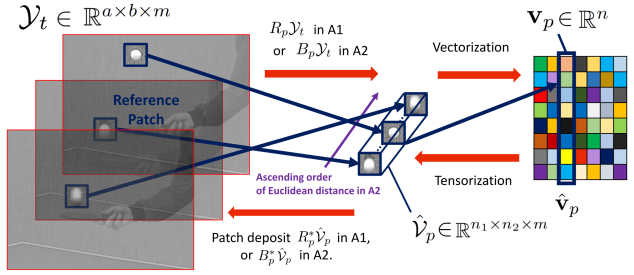


Fig. 5. Patch deposit $R_p^* \,\text{vec}^{-1}(\hat{\mathbf{v}}_p)$ (resp. $B_p^* \,\text{vec}^{-1}(\hat{\mathbf{v}}_p)$) as an adjoint of patch extraction operator in **A1** (resp. an adjoint of BM operator in **A2**).

B. VIDOSAT-BM

For videos with relatively static scenes, each extracted spatio-temporal tensor $R_p\mathcal{Y}_t$ in the VIDOSAT Algorithm **A1** typically has high temporal correlation, implying high (3D) transform domain sparsity. However, highly dynamic videos usually involve various motions, such as translation, rotation, scaling, etc. Figure 4 demonstrates one example when the 3D patch construction strategy in the VIDOSAT denoising algorithm **A1** fails to capture the properties of the moving object. Thus, Algorithm **A1** could provide sub-optimal denoising performance for highly dynamic videos. We propose an alternative algorithm, dubbed VIDOSAT-BM, which improves VIDOSAT denoising by constructing 3D patches using block matching.

The proposed VIDOSAT-BM solves the online transform learning problem (P3) with a different methodology for constructing the 3D patches and each mini-batch. The Steps $(2)-(4)$ in Algorithm **A1** remain the same for VIDOSAT-BM. We now discuss the modified Steps (1) and (5) in the VIDOSAT-BM denoising algorithm, to which we also refer as Algorithm **A2**.

1) 3D Patch and Mini-Batch Formation in VIDOSAT-BM: Here, we use a small and odd-valued sliding (temporal) window size m (e.g., we set $m = 9$ in the video denoising experiments in Section VI, which corresponds to ~ 0.2s buffer duration for a video with 40 Hz frame rate). Within the m-frame input FIFO buffer $\mathcal{Y}_t$, we approximate the various motions in the video using simple (local) translations [41].

We consider the middle frame $\mathbf{Y}_{t-(m-1)/2}$ in the input FIFO buffer $\mathcal{Y}_t$, and sequentially extract all 2D overlapping patches $\mathbf{Z}_p^t \in \mathbb{R}^{n_1 \times n_2}$, $1 \leq p \leq P$ in $\mathbf{Y}_{t-(m-1)/2}$, in a 2D spatially contiguous (raster scan) order. For each $\mathbf{Z}_p^t$, we form a $h_1 \times h_2 \times m$ pixel local search window centered at the center of $\mathbf{Z}_p^t$ (see the illustration in Fig. 4). We apply a spatial BM operator, denoted B_p, to find (using exhaustive search) the $(m-1)$ patches, one for each neighboring frame in the search window, that are most similar to $\mathbf{Z}_p^t$ in Euclidean distance. The operator B_p stacks the $\mathbf{Z}_p^t$, followed by the $(m-1)$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIP.2018.2865684, IEEE Transactions on Image Processing

Data	**ASU** Dataset (26 videos)					ΔPSNR	**LASIP** Dataset (8 videos)					ΔPSNR
σ	5	10	15	20	50	(std.)	5	10	15	20	50	(std.)
fBM3D [11]	38.78	34.66	32.38	30.82	26.13	3.89 (1.41)	38.05	34.06	31.89	30.42	25.88	2.11 (1.03)
fDnCNN [12]	N/A	32.59	30.78	30.90	26.43	4.82 (2.57)	N/A	35.05	33.01	31.55	26.81	1.12 (0.44)
sKSVD [5]	41.27	37.37	35.15	33.59	28.79	1.20 (0.34)	38.87	34.95	32.80	31.33	26.89	1.21 (0.38)
3D DCT	41.26	37.14	34.73	33.03	27.59	1.69 (0.78)	38.01	33.60	30.44	28.50	22.31	3.60 (1.28)
RNLF [18]	40.60	37.21	35.21	33.72	28.58	1.37 (0.29)	38.22	34.90	33.02	31.61	26.48	1.33 (0.36)
ReLD [21]	40.83	37.24	35.21	33.75	28.23	1.38 (0.31)	38.19	34.89	33.05	31.76	26.98	1.20 (0.35)
VBM3D [15]	41.10	37.82	35.78	34.25	28.65	0.92 (0.72)	39.20	35.75	33.87	32.49	26.51	0.61 (0.51)
VBM4D [19]	41.42	37.59	35.30	33.64	27.76	1.30 (0.86)	39.37	35.73	33.70	32.24	26.68	0.63 (0.49)
VIDOSAT	41.94	38.32	36.13	34.60	29.87	0.27 (0.13)	39.56	35.75	33.54	31.98	27.29	0.55 (0.29)
VIDOSAT -BM	**42.22**	**38.57**	**36.42**	**34.88**	**30.09**	0	**39.95**	**36.11**	**34.05**	**32.60**	**28.15**	0

TABLE III

COMPARISON OF VIDEO DENOISING PSNR VALUES (IN DB), AVERAGED TWO DATASETS FOR THE PROPOSED VIDOSAT, VIDOSAT-BM, AND OTHER COMPETING METHODS. FOR EACH DATASET AND NOISE LEVEL, THE BEST DENOISING PSNR IS MARKED IN BOLD. FOR EACH METHOD, WE LIST Δ PSNR, WHICH DENOTES THE AVERAGE PSNR LOSS (WITH ITS STANDARD DEVIATION IN PARENTHESES) RELATIVE TO THE PROPOSED VIDOSAT-BM.

matched patches, in an ascending order of their Euclidean distance to $\mathbf{Z}_p^t$, to form the pth 3D patch $B_p\mathcal{Y}_t \in \mathbb{R}^{n_1\times n_2\times m}$. Similar BM approaches have been used in prior works on video compression (e.g., MPEG) for motion compensation [41], and in recent works on spatiotemporal medical imaging [3]. The coordinates of all selected 2D patches are recorded to be used later in the denoised patch aggregation step. Instead of constructing the 3D patches from 2D patches in corresponding locations in contiguous frames (i.e., $R_p\mathcal{Y}_t$ in Algorithm A1), we form the patches using BM and work with the vectorized $\mathbf{v}_p^t = \mathrm{vec}(B_p\mathcal{Y}_t) \in \mathbb{R}^n$ in VIDOSAT-BM. The k-th mini-batch is defined as in Algorithm **A1** as $\mathbf{U}_{L_k^t} = \left[\mathbf{v}_{M(k-1)+1}^t \mid \ldots \mid \mathbf{v}_{Mk}^t\right]$.

2) Aggregation: Each denoised 3D patch (tensor) of $\left\{\mathrm{vec}^{-1}(\hat{\mathbf{v}}_p^t)\right\}_{p\in S_k}$ contains the matched (and denoised) 2D patches. They are sequentially aggregated at their recorded spatial and temporal locations in the output FIFO buffer $\bar{\mathcal{Y}}_t$ as $\sum_{p\in S_k} B_p^* \,\mathrm{vec}^{-1}(\hat{\mathbf{v}}_p^t) \rightarrow \bar{\mathcal{Y}}_t \in \mathbb{R}^{a\times b\times m}$, where the adjoint B_p^* is the patch deposit operator in **A2**. Fig. 5 illustrates the patch deposit procedure for aggregation in **A2**. Once the aggregation of $\bar{\mathcal{Y}}_t$ completes, the oldest frame in $\bar{\mathcal{Y}}_t$ is normalized pixel-wise by the number of occurrences of each pixel among patches in the denoising algorithm. Unlike Algorithm **A1** where this number of occurrences is the same for all frames, in Algorithm **A2** this number is data-dependent and varies from frame to frame and pixel to pixel. We record the number of occurrences of each pixel which is based on the recorded locations of the matched patches, and can be computed online as described. The normalized oldest frame is output by Algorithm **A2** for each time instant.

C. Computational Costs

In Algorithm **A1**, the computational cost of the sparse coding step is dominated by the computation of matrix-vector multiplication $\hat{\mathbf{W}}\mathbf{u}_i$, which scales as $O(Mn^2)$ [2, 7] for each mini-batch. The cost of mini-batch transform update step is $O(n^3 + Mn^2)$, which is dominated by full SVD and matrix-matrix multiplications. The cost of the 3D denoised patch reconstruction step also scales as $O(n^3+Mn^2)$ per mini-batch, which is dominated by the computation of matrix inverse $\hat{\mathbf{W}}^{-1}$ and multiplications. As all overlapping patches from a $a\times b\times T$ video are sequentially processed, the computational cost of Algorithm **A1** scales as $O(abTn^3/M + abTn^2)$. We set $M = 15n$ in practice, so that the cost of **A1** scales as $O(abTn^2)$. The cost of the additional BM step in Algorithm **A2** scales as $O(abTmh_1h_2)$, where $h_1 \times h_2$ is the

search window size. Therefore, the total cost of **A2** scales as $O(abTn^2 + abTmh_1h_2)$, which is on par with the state-of-the-art video denoising algorithm VBM3D [15], which is not an online method.

VI. Experiments

A. Implementation and Parameters

1) Testing Data: We present experimental results demonstrating the promise of the proposed VIDOSAT and VIDOSAT-BM online video denoising methods[5]. We evaluated the proposed algorithms by denoising all 34 videos from 2 public datasets, including 8 videos from the LASIP video dataset[6] [15, 19], and 26 videos of the Arizona State University (ASU) Video Trace Library[7] [42]. The testing videos contain 50 to 870 frames, with the frame resolution ranging from 176×144 to 720×576. Each video involves different types of motion, including translation, rotation, scaling (zooming), etc. The color videos are all converted to gray-scale. We simulated i.i.d. zero-mean Gaussian noise at 5 different noise levels (with noise standard deviation $\sigma = 5$, 10, 15, 20, and 50) for each video.

2) Implementation Details: We include several minor modifications of VIDOSAT and VIDOSAT-BM algorithms for improved performance. At each time instant t, we perform multiple passes of denoising for each $\mathcal{Y}_t$, by iterating over Steps (1) to (5) multiple times. In each pass, we denoise the output from the previous iteration [7, 27]. Given the noise standard deviation $\sigma_0 = \sigma$ of the initial input $\mathcal{Y}^0 = \mathcal{Y}$, in the jth pass, the σ_j of the noise remaining in the array $\bar{\mathcal{Y}}_t^{j-1}$ is re-estimated as $\sigma_j = \psi\sqrt{\sigma_0^2 - (1/abm)\|\bar{\mathcal{Y}}^{j-1} - \mathcal{Y}\|^2}$ [43, 44]. Here $(1/abm)\|\bar{\mathcal{Y}}^{j-1} - \mathcal{Y}\|^2$ and σ_j^2 approximate the variances of the noise removed and noise remaining, respecively, after the $j-1$ passes. The parameter $\psi = 0.6$ is set empirically (tuned over the training set) to best denoising performance. Applying multiple-pass denoising does not increase the inherent latency $m-1$ of the single-pass algorithm described earlier.

The following details are specifically for VIDOSAT-BM. First, instead of performing BM over the noisy input buffer $\mathcal{Y}_t$, we pre-clean $\mathcal{Y}_t$ using the VIDOSAT mini-batch denoising Algorithm **A1**, and then perform BM over the VIDOSAT denoised output. Second, when denoised 3D patches are aggregated to the output buffer, we assign them different weights, which are proportional to the sparsity level of their optimal sparse codes [45]. The weights are also accumulated and used for the output normalization.

3) Hyperparameters: We work with fully overlapping patches with spatial size $n_1 = n_2 = 8$, and temporal depth of $m = 9$ frames, which also corresponds to the depth of the buffer $\mathcal{Y}$. It follows that for a video with $N_1 \times N_2$ frames, the buffer $\mathcal{Y}$ contains mN_1N_2 pixels, and $P = (N_1 - n_1 + 1)(N_2 - n_2 + 1)$ 3D patches. We set the sparsity penalty weight parameter $\alpha_0 = 1.9$, the transform regularizer weight constant $\lambda_0 = 10^{-2}$, and the mini-batch size $M = 15 \times mn_1n_2$. The transform $\mathbf{W}$ is initialized at the beginning with the 3D DCT $\mathbf{W}_0$. For the other parameters, we adopt the settings in prior works [2, 7, 27], such as the forgetting factor $\rho = 0.68, 0.72, 0.76, 0.83, 0.89$, and the number of passes $L_p = 1, 2, 3, 3, 4$, for $\sigma = 5, 10, 15, 20$ and 50, respectively. The values of ρ and L_p both increase as the noise level increases. The larger ρ helps prevent overfitting to noise, and the larger number of passes improves denoising performance at higher noise level. For VIDOSAT-BM, we set the local search window size $h_1 = h_2 = 21$.

B. Video Denoising Results

1) Competing Methods: We compare the video denoising results obtained using the proposed VIDOSAT and VIDOSAT-BM algorithms to several well-known or recent alternatives, including the frame-wise BM3D (fBM3D) [11] and DnCNN (fDnCNN) [12] denoising methods, the image sequence denoising method using sparse KSVD (sKSVD) [5], the non-local means method (RNLF) [18], the method based on foreground and background separation (ReLD) [21], and the popular VBM3D [15] and VBM4D methods [19]. We used the publicly available implementations of these methods. For fDnCNN, we used the trained models that have been released by the authors [46] for $\sigma = 10$, 15, 20, and 50. Among these competing methods, fBM3D and fDnCNN denoise each frame independently by applying the popular BM3D and DnCNN image denoising methods, respectively; sKSVD exploits adaptive spatio-temporal sparsity but the dictionary is not learned online; RNLF denoises videos by applying non-local means filtering; ReLD separates the foreground and background of video frames, and denoises them by sparse and low-rank modeling, respectively; VBM3D and VBM4D are popular and state-of-the-art video denoising methods exploiting sparsity, block matching, and Wiener filtering. Moreover, to better understand the advantages of the adaptive online high-dimensional transform learning, we apply the proposed video denoising framework, but fixing the sparsifying transform in VIDOSAT to 3D DCT, which is referred as the 3D DCT method in the results.

2) Denoising Results: We present video denoising results using the proposed VIDOSAT and VIDOSAT-BM algorithms, as well as using the other aforementioned competing methods. To evaluate the performance of the various denoising schemes, we measured the video (3D) peak signal-to-noise ratio (PSNR) in decibels (dB), which is computed as the ratio of the peak intensity value of the noiseless reference video to the root mean square error between the reference and denoised videos. We also report the PSNRs computed for individual 2D frames (i.e., frame-wise PSNRs) of example videos, which are plotted against the frame numbers.

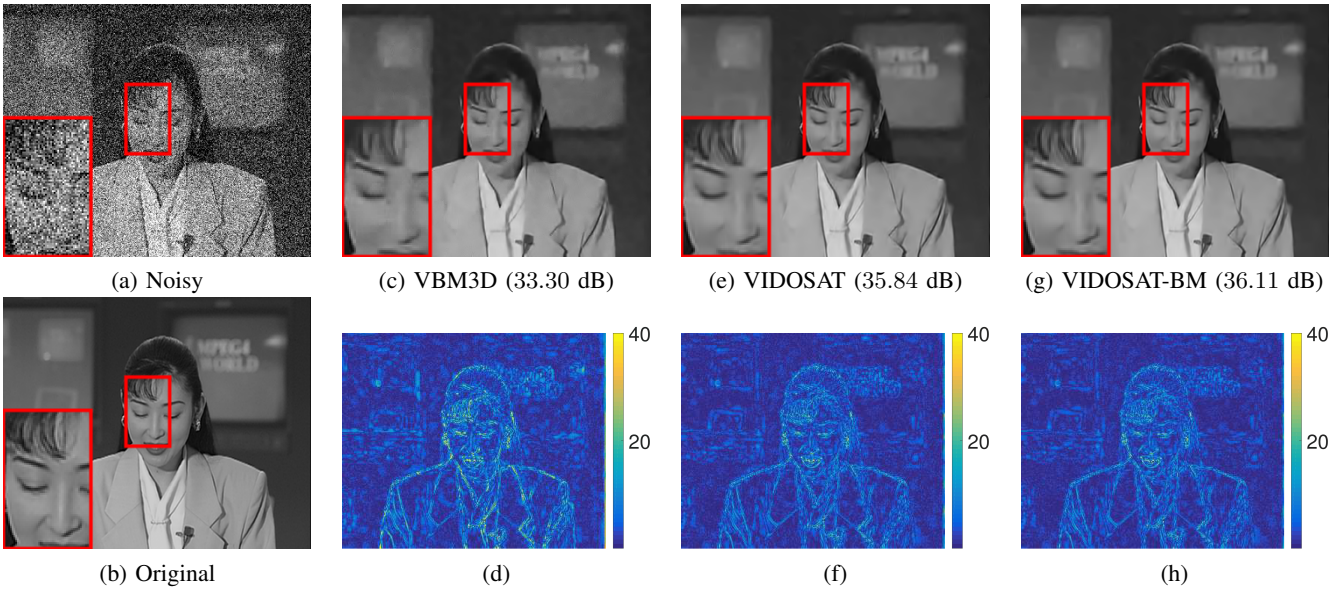
Table III lists the video denoising PSNRs obtained by the two proposed VIDOSAT methods as well as the eight competing methods. For each method, we list ΔPSNR, which denotes the average PSNR loss (with its standard deviation included in parentheses) relative to the proposed VIDOSAT-BM. As the DnCNN model for $\sigma = 5$ is unavailable (shown as N/A in Table III), the ΔPSNR (and its standard deviation)

[5]Matlab implementations of VIDOSAT and VIDOSAT-BM are publicly available at http://transformlearning.csl.illinois.edu.

[6]Available at http://www.cs.tut.fi/~lasip/foi_wwwstorage/test_videos.zip

[7]Available at http://trace.eas.asu.edu/yuv/. Only videos with less than 1000 frames are selected for our image denoising experiments.

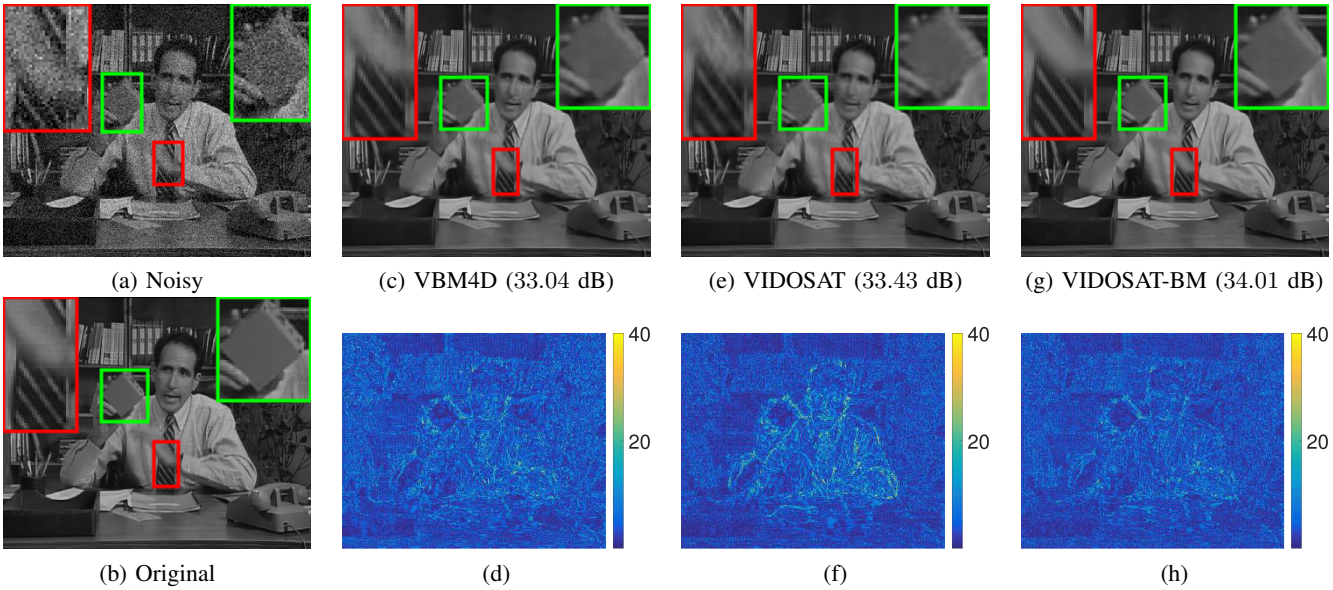
This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIP.2018.2865684, IEEE Transactions on Image Processing



(a) Noisy (c) VBM3D (33.30 dB) (e) VIDOSAT (35.84 dB) (g) VIDOSAT-BM (36.11 dB)

(b) Original (d) (f) (h)

Fig. 6. (a) The noisy version ($\sigma = 50$) of (b) one frame of the *Akiyo* ($288 \times 352 \times 300$) video. We show the comparison of the denoising results (resp. the magnitude of error in the denoised frame) using (c) VBM3D (33.30 dB), (e) VIDOSAT (35.84 dB) and (g) VIDOSAT-BM (36.11 dB) (resp. (d), (f) and (h)). The PSNR of the denoised frame is shown in the parentheses. The zoom-in region is highlighted using red box.



(a) Noisy (c) VBM4D (33.04 dB) (e) VIDOSAT (33.43 dB) (g) VIDOSAT-BM (34.01 dB)

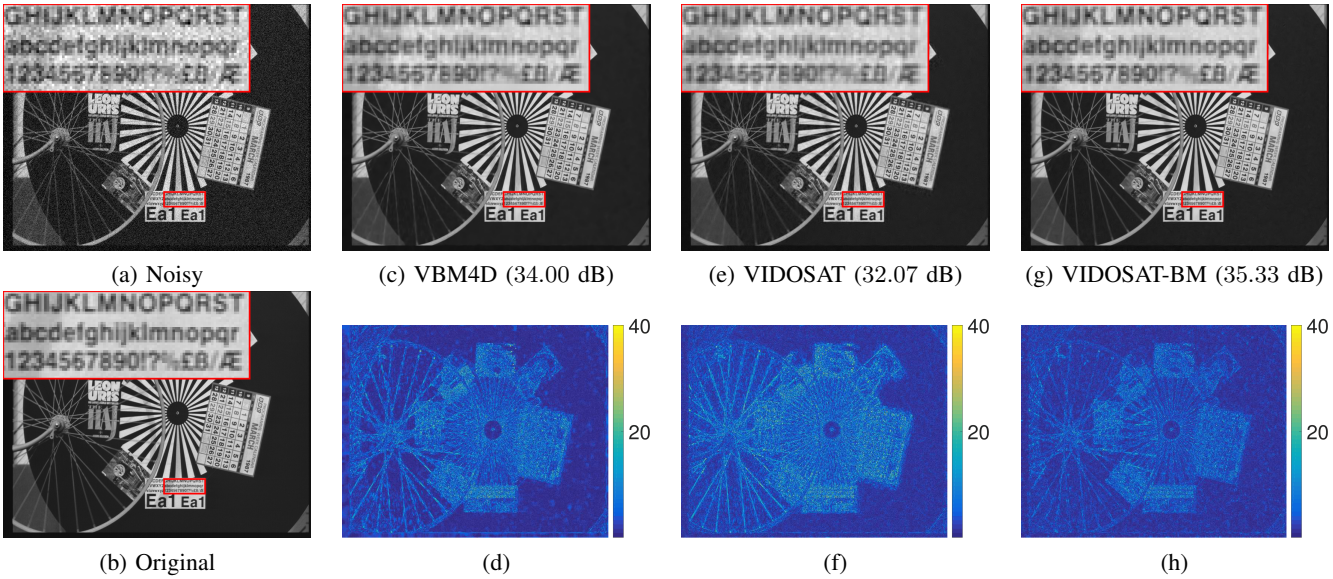
(b) Original (d) (f) (h)

Fig. 7. (a) The noisy version ($\sigma = 20$) of (b) one frame of the *Salesman* ($288 \times 352 \times 50$) video. We show the comparison of the denoising results (resp. the magnitude of error in the denoised frame) using (c) VBM4D (33.04 dB), (e) VIDOSAT (33.43 dB) and (g) VIDOSAT-BM (34.01 dB) (resp. (d), (f) and (h)). The PSNR of the denoised frame is shown in the parentheses. The zoom-in regions are highlighted using red and green boxes.

using fDnCNN is computed excluding $\sigma = 5$. It is clear that the proposed VIDOSAT and VIDOSAT-BM methods both generate better denoising results with higher average PSNR values compared to the competing methods. The VIDOSAT-BM denoising method provides average PSNR improvements (averaged over all 34 testing videos from both datasets and all noise levels) of 0.9 dB, 1.1 dB, 1.2 dB, 1.3 dB, 1.4 dB, 2.1 dB, 3.5 dB, and 3.9 dB over the VBM3D, VBM4D, sKSVD, ReLD, RNLF, 3D DCT, fBM3D, and fDnCNN denoising methods, respectively. Importantly, for each testing video and noise level, VIDOSAT-BM was observed to consistently outperform all the competing methods. Among the two proposed VIDOSAT algorithms, the average video denoising PSNR achieved by VIDOSAT-BM is 0.3 dB higher than that using the VIDOSAT method, owing to the effectiveness of block matching for modeling dynamics and motion in videos. Fig. 10 plots the denoised PSNRs of videos *Salesman* and *Bicycle* for various temporal depths m and $\sigma = 20$ using VIDOSAT-BM. When using larger m, the VIDOSAT-BM algorithm exploits video correlation with longer temporal range, while BM becomes less accurate due to noise corruption. Thus, there is an optimal m observed for each video and noise level. We observed that the chosen $m = 9$ works well across our experiments. Note that large m values also increase the latency and computational cost of the algorithms.

Next, we illustrate some denoised results using VIDOSAT and VIDOSAT-BM. To demonstrate the visual quality improvements achieved by the proposed methods, we also show

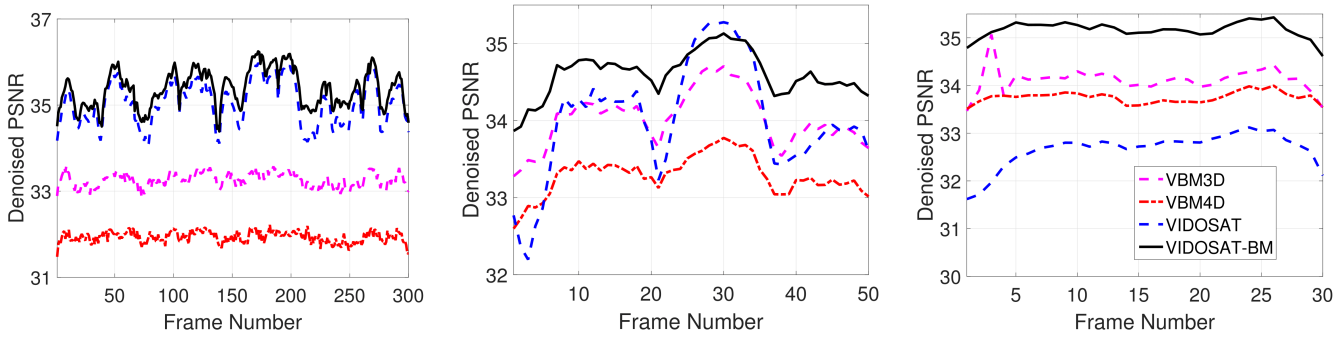
This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIP.2018.2865684, IEEE Transactions on Image Processing



(a) Noisy (c) VBM4D (34.00 dB) (e) VIDOSAT (32.07 dB) (g) VIDOSAT-BM (35.33 dB)

(b) Original (d) (f) (h)

Fig. 8. (a) The noisy version ($\sigma = 20$) of (b) one frame of the *Bicycle* ($576 \times 720 \times 30$) video. We show the comparison of the denoising results (resp. the magnitude of error in the denoised frame) using (c) VBM4D (34.00 dB), (e) VIDOSAT (32.07 dB) and (g) VIDOSAT-BM (35.33 dB) (resp. (d), (f) and (h)). The PSNR of the denoised frame is shown in the parentheses. The zoom-in region is highlighted using red box.



(a) (b) (c)

Fig. 9. Frame-by-frame PSNR (dB) for (a) *Akiyo* with $\sigma = 50$, (b) *Salesman* with $\sigma = 20$, and (c) *Bicycle* with $\sigma = 20$, denoised by VBM3D, VBM4D, and the proposed VIDOSAT and VIDOSAT-BM schemes, respectively.

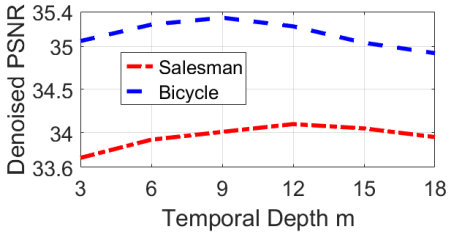


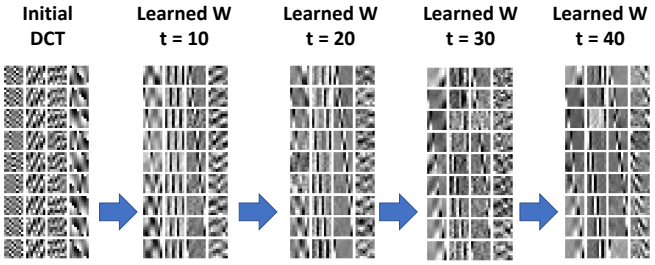
Fig. 10. Plot of denoised video PSNRs by VIDOSAT-BM for several temporal depths m for *Salesman* and *Bicycle* with $\sigma = 20$.

the results obtained by VBM3D and VBM4D, which are the best algorithms among the competing methods in Table III.

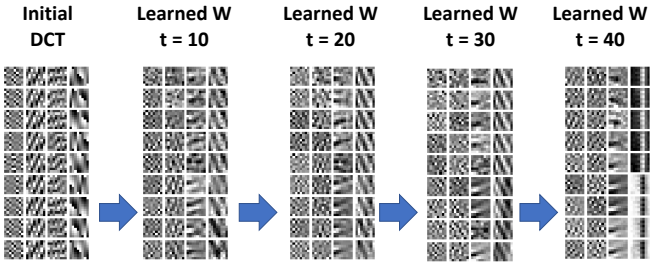
a) Fig. 6 shows one denoised frame of the video *Akiyo* ($\sigma = 50$), which involves static background and a relatively small moving region. The magnitudes of the denoising errors in Fig. 6 are clipped for viewing. The denoising results by VIDOSAT and VIDOSAT-BM both demonstrate similar visual quality improvements over the result by VBM3D. Fig. 9(a) shows the frame-by-frame PSNRs of the denoised *Akiyo*, in which VIDOSAT and VIDOSAT-BM provide comparable denoising PSNRs, and both outperform the VBM3D and VBM4D schemes consistently by a sizable margin.

b) Fig. 7 shows one denoised frame of the video *Salesman* ($\sigma = 20$) that involves occasional but fast movements (e.g., hand waving) in the foreground. The denoising result by VIDOSAT improves over the VBM4D result in general, but also shows some artifacts in regions with strong motion. Instead, the result by VIDOSAT-BM provides the best visual quality in both the static and the moving parts. Fig. 9(b) shows the frame-by-frame PSNRs of the denoised *Salesman*. VIDOSAT-BM provides large improvements over the other methods including VIDOSAT for most frames, and the PSNR is more stable (smaller deviations) over frames. Fig. 11 shows example atoms (i.e., rows) of the initial 3D DCT transform, and the online learned transforms using VIDOSAT and VIDOSAT-BM, at different times t. For the learned $\hat{\mathbf{W}}_t$'s using both VIDOSAT and VIDOSAT-BM, their atoms are observed to gradually evolve, in order to adapt to the dynamic video content. The learned transform atoms using VIDOSAT in Fig. 11(a) demonstrate linear shifting structure along the patch depth m, which is likely to compensate the video motion (e.g., translation). On the other hand, since the 3D patches are formed using block matching in VIDOSAT-BM, such structure is not observed in Fig. 11(b) when $\hat{\mathbf{W}}_t$ is learned using

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIP.2018.2865684, IEEE Transactions on Image Processing



(a) $\hat{\mathbf{W}}_t$ learned using VIDOSAT



(b) $\hat{\mathbf{W}}_t$ learned using VIDOSAT-BM

Fig. 11. Example atoms (i.e., 4 rows) of the initial 3D DCT (with depth $m = 9$), and the online learned 3D sparsifying transform using (a) VIDOSAT, and (b) VIDOSAT-BM, at times 10 to 40 (learned at same spatial location): the atoms (i.e., rows) of the learned $\hat{\mathbf{W}}$ are shown as $m = 9$ patches in each column above. These 9 patches together form an $8 \times 8 \times 9$ 3D atom.

VIDOSAT-BM.

c) Fig. 8 shows one denoised frame of the video *Bicycle* ($\sigma = 20$), which contains a large area of complex movements (e.g., rotations) throughout the video. In this case, the denoised frame using VIDOSAT is worse than VBM4D. However, VIDOSAT-BM provides superior quality compared to all the methods. This example demonstrates the effectiveness of joint block matching and learning in the proposed VIDOSAT-BM scheme, especially when processing highly dynamic videos. Fig. 9(c) shows the frame-by-frame PSNRs of the denoised *Bicycle*, in which VIDOSAT-BM significantly improves over VIDOSAT, and also outperforms both VBM3D and VBM4D for all frames.

VII. Conclusions

We presented a novel framework for online video denoising based on efficient high-dimensional sparsifying transform learning. The transforms are learned in an online manner from appropriately constructed 3D (spatio-temporal) patches. These patches are constructed either from corresponding 2D patches of consecutive frames or using an online block matching technique. The learned models effectively capture the dynamic changes in videos. We demonstrated the promising performance of the proposed video denoising schemes for several standard datasets. Our methods outperformed all compared methods, which included a version of the proposed video denoising scheme in which the learning of the sparsifying transform was eliminated and instead it was fixed to 3D DCT, as well as denoising using learned synthesis dictionaries, deep neural networks, the recent ReLD and RNLF, and the state-of-the-art VBM3D and VBM4D methods. While this work provides an initial study of the promise of the proposed data-driven online video denoising methodologies, we leave the study of the potential implementation and acceleration of the proposed schemes for real-time video processing and extensions to include richer overcomplete transforms [27, 36, 47] to future work.

References

[1] N. Rajpoot, Z. Yao, and R. Wilson, "Adaptive wavelet restoration of noisy video sequences," in *IEEE International Conference on Image Processing (ICIP)*, vol. 2, 2004, pp. 957–960.

[2] B. Wen, S. Ravishankar, and Y. Bresler, "Video denoising by online 3D sparsifying transform learning," in *IEEE International Conference on Image Processing (ICIP)*, 2015, pp. 118–122.

[3] H. Yoon, K. S. Kim, D. Kim, Y. Bresler, and J. C. Ye, "Motion adaptive patch-based low-rank approach for compressed sensing cardiac cine MRI," *IEEE Transactions on Medical Imaging*, vol. 33, no. 11, pp. 2069–2085, 2014.

[4] M. Protter and M. Elad, "Image sequence denoising via sparse and redundant representations," *IEEE Transactions on Image Processing*, vol. 18, no. 1, pp. 27–35, 2009.

[5] 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.

[6] S. Ravishankar and Y. Bresler, "Sparsifying transform learning with efficient optimal updates and convergence guarantees," *IEEE Transactions on Signal Processing*, vol. 63, no. 9, pp. 2389–2404, 2015.

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

[8] B. Wen, Y. Li, L. Pfister, and Y. Bresler, "Joint adaptive sparsity and low-rankness on the fly: an online tensor reconstruction scheme for video denoising," in *IEEE International Conference on Computer Vision (ICCV)*, 2017.

[9] D. Liu, B. Wen, X. Liu, and T. S. Huang, "When image denoising meets high-level vision tasks: A deep learning approach," in *International Joint Conference on Artificial Intelligence (IJCAI)*, 2018.

[10] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, and P.-A. Manzagol, "Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion," *Journal of Machine Learning Research*, vol. 11, no. Dec, pp. 3371–3408, 2010.

[11] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian, "Image denoising by sparse 3-D transform-domain collaborative filtering," *IEEE Transactions on Image Processing*, vol. 16, no. 8, pp. 2080–2095, 2007.

[12] K. Zhang, W. Zuo, Y. Chen, D. Meng, and L. Zhang, "Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising," *IEEE Transactions on Image Processing*, vol. 26, no. 7, pp. 3142–3155, 2017.

[13] R. Otazo, E. Candès, and D. K. Sodickson, "Low-rank plus sparse matrix decomposition for accelerated dynamic MRI with separation of background and dynamic components," *Magnetic Resonance in Medicine*, vol. 73, no. 3, pp. 1125–1136, 2015.

[14] 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.

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

[16] P. Arias and J.-M. Morel, "Video denoising via empirical bayesian estimation of space-time patches," *Journal of Mathematical Imaging and Vision*, vol. 60, no. 1, pp. 70–93, 2018.

[17] A. Buades, J.-L. Lisani, and M. Miladinović, "Patch-based video denoising with optical flow estimation," *IEEE Transactions on Image Processing*, vol. 25, no. 6, pp. 2573–2586, 2016.

[18] C. Sutour, C.-A. Deledalle, and J.-F. Aujol, "Adaptive regularization of the nl-means: Application to image and video denoising," *IEEE Transactions on Image Processing*, vol. 23, no. 8, pp. 3506–3521, 2014.

[19] M. Maggioni, G. Boracchi, A. Foi, and K. Egiazarian, "Video denoising, deblocking, and enhancement through separable 4-d nonlocal spatiotemporal transforms," *IEEE Transactions on Image Processing*, vol. 21, no. 9, pp. 3952–3966, 2012. [Online]. Available: http://dx.doi.org/10.1109/TIP.2012.2199324

[20] H. Ji, S. Huang, Z. Shen, and Y. 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.

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

[22] M. Elad and M. Aharon, "Image denoising via sparse and redundant representations over learned dictionaries," *IEEE Transactions on Image Processing*, vol. 15, no. 12, pp. 3736–45, Dec. 2006.

[23] J. Mairal, M. Elad, and G. Sapiro, "Sparse representation for color image restoration," *IEEE Trans. on Image Processing*, vol. 17, no. 1, pp. 53–69, 2008.

[24] I. Ramirez, P. Sprechmann, and G. Sapiro, "Classification and clustering via dictionary learning with structured incoherence and shared features," in *Proc. IEEE International Conference on Computer Vision and Pattern Recognition (CVPR) 2010*, 2010, pp. 3501–3508.

[25] S. Ravishankar and Y. Bresler, "MR image reconstruction from highly undersampled k-space data by dictionary learning," *IEEE Trans. Med. Imag.*, vol. 30, no. 5, pp. 1028–1041, 2011.

[26] R. Rubinstein, T. Peleg, and M. Elad, "Analysis K-SVD: A dictionary-learning algorithm for the analysis sparse model," *IEEE Transactions on Signal Processing*, vol. 61, no. 3, pp. 661–677, 2013.

[27] B. Wen, S. Ravishankar, and Y. Bresler, "Structured overcomplete sparsifying transform learning with convergence guarantees and applications," *International Journal of Computer Vision*, vol. 114, no. 2-3, pp. 137–167, 2015.

[28] J.-F. Cai, H. Ji, Z. Shen, and G.-B. Ye, "Data-driven tight frame construction and image denoising," *Applied and Computational Harmonic Analysis*, vol. 37, no. 1, pp. 89 – 105, 2014.

[29] Z. Zhan, J.-F. Cai, D. Guo, Y. Liu, Z. Chen, and X. Qu, "Fast multiclass dictionaries learning with geometrical directions in mri reconstruction," *IEEE Transactions on Biomedical Engineering*, vol. 63, no. 9, pp. 1850–1861, 2016.

[30] X. Zheng, Z. Lu, S. Ravishankar, Y. Long, and J. A. Fessier, "Low dose CT image reconstruction with learned sparsifying transform," in *Proc. IEEE Wkshp. on Image, Video, Multidim. Signal Proc.*, Jul. 2016, pp. 1–5.

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

[32] Y. C. Pati, R. Rezaiifar, and P. S. Krishnaprasad, "Orthogonal matching pursuit : recursive function approximation with applications to wavelet decomposition," in *Asilomar Conf. on Signals, Systems and Comput.*, 1993, pp. 40–44 vol.1.

[33] B. Efron, T. Hastie, I. Johnstone, and R. Tibshirani, "Least angle regression," *Annals of Statistics*, vol. 32, pp. 407–499, 2004.

[34] D. Needell and J. Tropp, "CoSaMP: Iterative signal recovery from incomplete and inaccurate samples," *Applied and Computational Harmonic Analysis*, vol. 26, no. 3, pp. 301–321, 2009.

[35] W. Dai and O. Milenkovic, "Subspace pursuit for compressive sensing signal reconstruction," *IEEE Trans. Information Theory*, vol. 55, no. 5, pp. 2230–2249, 2009.

[36] B. Wen, S. Ravishankar, and Y. Bresler, "FRIST — flipping and rotation invariant sparsifying transform learning and applications," *Inverse Problems*, vol. 33, no. 7, p. 074007, 2017.

[37] L. Pfister and Y. Bresler, "Model-based iterative tomographic reconstruction with adaptive sparsifying transforms," in *Proc. SPIE Computational Imaging XII*, C. A. Bouman and K. D. Sauer, Eds. SPIE, Mar. 2014, pp. 90 200H–90 200H–11.

[38] J. Mairal, F. Bach, J. Ponce, and G. Sapiro, "Online learning for matrix factorization and sparse coding," *Journal of Machine Learning Research*, vol. 11, no. Jan, pp. 19–60, 2010.

[39] T. Ouni and M. Abid, "Scan methods and their application in image compression," *International Journal of Signal Processing, Image Processing and Pattern Recognition*, vol. 5, no. 3, pp. 49–64, 2012.

[40] S. Mallat, *A wavelet tour of signal processing*. Orlando, FL, USA: Academic Press, 1999.

[41] D. Le Gall, "MPEG: A video compression standard for multimedia applications," *Communications of the ACM*, vol. 34, no. 4, pp. 46–58, 1991.

[42] P. Seeling and M. Reisslein, "Video traffic characteristics of modern encoding standards: H. 264/avc with svc and mvc extensions and h. 265/hevc," *The Scientific World Journal*, vol. 2014, 2014.

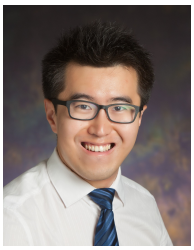
[43] S. Gu, Q. Xie, D. Meng, W. Zuo, X. Feng, and L. Zhang, "Weighted nuclear norm minimization and its applications to low level vision," *International Journal of Computer Vision*, vol. 121, no. 2, pp. 183–208, 2017.

[44] Y. Romano and M. Elad, "Boosting of image denoising algorithms," *SIAM Journal on Imaging Sciences*, vol. 8, no. 2, pp. 1187–1219, 2015.

[45] B. Wen, Y. Li, and Y. Bresler, "When sparsity meets low-rankness: transform learning with non-local low-rank constraint for image restoration," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2017, pp. 2297–2301.

[46] K. Zhang, W. Zuo, Y. Chen, D. Meng, and L. Zhang, "DnCNN," https://github.com/cszn/DnCNN, 2017, gitHub repository.

[47] S. Ravishankar and Y. Bresler, "Learning overcomplete sparsifying transforms for signal processing," in *Int. Conf. Acoustics, Speech and Signal Proc., 2013*. IEEE, 2013, pp. 3088–3092.



Bihan Wen received the B.Eng. degree in electrical and electronic engineering from Nanyang Technological University, Singapore, in 2012, the M.S and Ph.D. degrees in electrical and computer engineering from the University of Illinois at Urbana-Champaign, in 2015 and 2018, respectively. He was the recipient of the 2016 Yee Fellowship, and the 2012 Professional Engineers Board Gold Medal. He won the 1st place in the 3MT Thesis Competition at ICME 2018, and the Best Talk Award at CSLSC 2018. His work was awarded 10% Best Paper in ICIP 2014. His research interests span areas of machine learning, computer vision, image and video processing, compressed sensing, and big data applications.

Saiprasad Ravishankar received the B.Tech. degree in Electrical Engineering from the Indian Institute of Technology Madras, in 2008. He received the M.S. and Ph.D. degrees in Electrical and Computer Engineering, in 2010 and 2014 respectively, from the University of Illinois at Urbana-Champaign, where he was an Adjunct Lecturer in the Department of Electrical and Computer Engineering during Spring 2015, and a Postdoctoral Research Associate at the Coordinated Science Laboratory until August, 2015. After that he was a Research Fellow in the Electrical Engineering and Computer Science Department at the University of Michigan. His current research interests include signal and image processing, biomedical and computational imaging, data-driven methods, machine learning, signal modeling, data science, dictionary learning, inverse problems, compressed sensing, and large-scale data processing. He was awarded the IEEE Signal Processing Society Young Author Best Paper Award for 2016.

Yoram Bresler received the B.Sc. (cum laude) and M.Sc. degrees from the Technion, Israel Institute of Technology, in 1974 and 1981, respectively, and the Ph.D. degree from Stanford University, Stanford, CA, USA, in 1986, all in electrical engineering. In 1987 he joined the University of Illinois at Urbana-Champaign, where he currently holds the title of a Founder Professor of Engineering with the Departments of Electrical and Computer Engineering and Bioengineering, and the Coordinated Science Laboratory. He is also President and Chief Technology Officer at InstaRecon, Inc., a startup company he cofounded to commercialize breakthrough technology for tomographic reconstruction developed in his academic research. His current research interests include machine learning and statistical signal processing and their applications to inverse problems in imaging, and in particular compressed sensing, computed tomography, and magnetic resonance imaging.

Dr. Bresler has served on the editorial board of a number of journals including the IEEE TRANSACTIONS ON SIGNAL PROCESSING, the IEEE JOURNAL ON SELECTED TOPICS IN SIGNAL PROCESSING, Machine Vision and Applications, and the SIAM Journal on Imaging Science, and on various committees of the IEEE. He is an IEEE Fellow, and Fellow of the AIMBE. He received two Senior Paper Awards from the IEEE Signal Processing society, and two papers he coauthored with his students received the Young Author Award from the same society in 2002 and 2016. He is the recipient of a 1991 NSF Presidential Young Investigator Award, the Technion Fellowship in 1995, and the Xerox Senior Award for Faculty Research in 1998. He was named a University of Illinois Scholar in 1999, appointed as an Associate at the Center for Advanced Study of the University in 2001–2002, and Faculty Fellow at the National Center for Super Computing Applications in 2006. In 2016 he was appointed an IEEE Signal Processing Society Distinguished Lecturer.