Online Adaptive Image Reconstruction (OnAIR) Using Dictionary Models

Brian E. Moore ⬤, *Member, IEEE*, Saiprasad Ravishankar ⬤, *Member, IEEE*, Raj Rao Nadakuditi, *Member, IEEE*, and Jeffrey A. Fessler ⬤, *Fellow, IEEE*

Abstract—**Sparsity and low-rank models have been popular for reconstructing images and videos from limited or corrupted measurements. Dictionary or transform learning methods are useful in applications such as denoising, inpainting, and medical image reconstruction. In this paper, we propose a framework for online (or time-sequential) adaptive reconstruction of dynamic image sequences from linear (typically undersampled) measurements. We model the spatiotemporal patches of the underlying dynamic image sequence as sparse in a dictionary, and we simultaneously estimate the dictionary and the images sequentially from streaming measurements. Multiple constraints on the adapted dictionary are also considered such as a unitary matrix, or low-rank dictionary atoms that provide additional efficiency or robustness. The proposed online algorithms are memory efficient and involve simple updates of the dictionary atoms, sparse coefficients, and images. Numerical experiments demonstrate the usefulness of the proposed methods in inverse problems such as video reconstruction or inpainting from noisy, subsampled pixels, and dynamic magnetic resonance image reconstruction from very limited measurements.**

Index Terms—**Online methods, sparse representations, dictionary learning, machine learning, inverse problems, video processing, dynamic magnetic resonance imaging.**

I. INTRODUCTION

MODELS of signals and images based on sparsity, low-rank, and other properties are useful in image and video processing. In ill-posed or ill-conditioned inverse problems, it is often useful to employ signal models that reflect known or assumed properties of the latent images. Such models are often used to construct appropriate regularization. For example, the sparsity of images in wavelet or discrete cosine transform (DCT) domains has been exploited for image and video reconstruction tasks [1]–[3]. In particular, the learning of such models has been explored in various settings [4]–[10], where they may potentially outperform fixed models since they adapt to signals and signal classes.

There has been growing interest in such dictionary learning-based image restoration or reconstruction methods [11]–[13]. For example, in blind compressed sensing [10], [12], [13], a dictionary for the underlying image or video is estimated together with the image from undersampled measurements. This allows the dictionary to adapt to the current data, which may enable learning novel features, providing improved reconstructions.

For inverse problems involving large-scale or streaming data, e.g., in interventional imaging or restoring (e.g, denoising, inpainting) large or streaming videos, etc., it is often critical to obtain reconstructions in an online or time-sequential (or data-sequential) manner to limit latency. Batch methods that process all the data at once are often prohibitively expensive in terms of time and memory usage, and infeasible for streaming data. Methods for online learning of dictionary and sparsifying transform models from streaming signals including noisy data have been recently proposed [14]–[18] and shown to outperform state-of-the-art methods [19]–[21] including deep learning methods for video denoising.

This paper focuses on methods for dynamic image reconstruction from limited measurements such as video or medical imaging data. One such important class of dynamic image reconstruction problems arises in dynamic magnetic resonance imaging (dMRI), where the data are inherently or naturally undersampled because the object is changing as the data (samples in k-space or Fourier space of the object acquired sequentially over time) is collected. Various techniques have been proposed for reconstructing such dynamic image sequences from limited measurements [22]–[24]. Such methods may achieve improved spatial or temporal resolution by using more explicit signal models compared to conventional approaches (such as k-space data sharing in dMRI, where data is pooled in time to make sets of k-space data with sufficient samples [25]); these methods typically achieve increased accuracy at the price of increased computation.

Recently there has been substantial interest in developing real-time dMRI reconstruction algorithms. This literature generally falls into one (or both) of the following categories [26]: methods that model the measurements as a time sequential or adaptive process, and those that generate reconstructions with minimal physical latency and are suited for real-time use in practice. Although the second of these categories is interesting

and important, methods with online or time sequential properties are valuable in their own right as they often provide benefits such as adaptability to time-varying aspects of the data and reduced memory usage and computation requirements.

While some reconstruction techniques are driven by sparsity models and assume that the image sequence is sparse in some transform domain or dictionary [22], other methods exploit low-rank or other kinds of sophisticated models [24], [25], [27]–[31]. For example, L+S methods [24], [31] assume that the image sequence can be decomposed as the sum of low-rank and sparse (either directly sparse or sparse in some known transform domain) components that are estimated from measurements. Dictionary learning-based approaches including with low-rank models provide promising performance for dynamic image reconstruction [12], [32], [33]. Although these methods allow adaptivity to data and provide improved reconstructions, they involve batch processing and typically expensive computation and memory use. Next, we outline our contributions, particularly a new online framework that address these issues of state-of-the-art batch reconstruction methods.

A. Contributions

This paper investigates a framework for Online Adaptive Image Reconstruction, dubbed OnAIR, exploiting learned dictionaries. We model spatiotemporal patches of the underlying dynamic image sequence as sparse in an (unknown) dictionary, and we propose a method to jointly and sequentially (over time) estimate the dictionary, sparse codes, and images from streaming measurements. Various constraints are considered for the dictionary model such as a unitary matrix, or a matrix whose atoms/columns are low-rank when reshaped into spatio-temporal matrices. The proposed OnAIR algorithms involve simple and efficient updates and require a small, fixed amount of data to be stored in memory at a time. Our method allows for time-varying image models that can adapt to temporal dynamics of the underlying data. Note that we use the term "online" in this work as it is used in the signal processing literature [34], not to imply that it is designed to process streaming measurements with sub-real-time latency. That is, our method is online in the sense that it processes measurements in a time sequential manner and can be interpreted in the empirical risk minimization framework [35].

Numerical experiments demonstrate the effectiveness of the proposed OnAIR methods for performing video inpainting from subsampled and noisy pixels, and dynamic MRI reconstruction from very limited k-t space measurements. The experiments show that our proposed methods are able to learn dictionaries adaptively from corrupted measurements with important representational features that improve the quality of the reconstructions compared to non-adaptive schemes. Moreover, the OnAIR methods provide better image quality than batch schemes with a single adapted dictionary, and with much lower runtimes than adaptive batch (or offline) reconstructions.

While dictionary learning was studied for image reconstruction in [33] and earlier works, the methods involved batch processing, where the model is learned based on patches from the entire dynamic dataset concurrently. Online dictionary learning is an important and well-studied problem in the literature as well [14]–[18]; however, this work is the first to study online dictionary learning-driven reconstruction from limited dynamic imaging data. Compared to previous batch reconstruction methods, the proposed OnAIR methods enjoy a number of advantages. First, the proposed methods involve modest memory use independent of the number of frames processed, while batch learning methods such as in [33] have a high memory requirement because they process the entire dataset and its patches synchronously, which is infeasible for large datasets. In addition, our proposed OnAIR methods use warm start initializations and sliding windows that make the method computationally efficient, requiring only a few iterations per minibatch to update the solution/model based on new data. On the other hand, batch methods such as [33] require many iterations over the entire dataset to ensure convergence. As a result, OnAIR enjoys significant speedups over the batch method in [33] as discussed in Section V. The OnAIR methods also allow for dictionaries that evolve over time, thereby adapting to the dynamic changes in the object. The experiments in Section V show that such online learned dictionaries better model the data and outperform the batch method from [33] in terms of image quality. Finally, the OnAIR methods offer reduced latency compared to batch methods such as [33] because the data can be processed and reconstructed sequentially over time without waiting until the entire dataset is measured, which is infeasible for applications involving streaming data.[1]

Short versions of this work appeared recently in [36] and [37]. This paper extends those initial works by exploring OnAIR methods with multiple constraints on the learned models including a Unitary Dictionary (OnAIR-UD) constraint, and a Low-rank Dictionary atom (OnAIR-LD) constraint. Such constraints may offer a degree of algorithmic efficiency and robustness to artifacts in limited data settings. For example, the updates per time instant in OnAIR-UD are simpler and non-sequential, enabling it to be faster in practice than the other OnAIR schemes. Importantly, this paper reports extensive numerical experiments investigating and evaluating the performance of the proposed online methods in multiple inverse problems, and comparing their performance against recent related online and batch methods in the literature. Finally, we also compare the performance of the OnAIR methods to an oracle online scheme, where the dictionary is learned offline from the ground truth data, and show that the proposed online learning from highly limited and corrupted streaming measurements performs comparably to the oracle scheme.

B. Organization

The rest of this paper is organized as follows. Section II reviews the dictionary learning framework that forms the basis of

---

[1]We also note that [33] includes a model called LASSI that incorporates a global low-rank property along with the dictionary learning model. This method was shown to provide small improvements over DINO-KAT (a dictionary learning-only method) in Tables I-III of [33]. Extending the OnAIR framework presented here to include such global low-rank models or other priors is an interesting problem that we leave to future work.

our online schemes. Section III describes our formulations for online adaptive image reconstruction and Section IV presents algorithms for solving the problems. Section V presents extensive numerical experiments that demonstrate the promising performance of our proposed methods in inverse problem settings such as video inpainting and dynamic MRI reconstruction from highly limited data. Finally, Section VI concludes with proposals for future work.

II. DICTIONARY LEARNING MODELS

Here, we briefly review some prior works on dictionary learning that helps build our OnAIR framework. Given a set of signals (or vectorized image patches) that are represented as columns of a matrix $P \in \mathbb{C}^{n \times M}$, the goal of dictionary learning (DL) is to learn a dictionary $D \in \mathbb{C}^{n \times m}$ and a matrix $Z \in \mathbb{C}^{m \times M}$ of sparse codes such that $P \approx DZ$. Traditionally, the DL problem is often formulated [4] as follows:

$$\min_{D,Z} \|P - DZ\|_F^2$$

$$\text{s.t. } \|d_i\|_2 = 1, \|z_l\|_0 \leq s, \forall i, l, \quad (1)$$

where d_i and z_l denote the ith column of D and the lth column of Z, respectively, and s denotes a target sparsity level for each signal. Here, the ℓ_0 "norm" counts the number of non-zero entries of its argument, and the columns of D are set to unit norm to avoid scaling ambiguity between D and Z [38]. Alternatives to (1) exist that replace the ℓ_0 "norm" constraint with other sparsity-promoting constraints, or enforce additional properties on the dictionary [39]–[41], or enable online dictionary learning [14].

Dictionary learning algorithms [4], [5], [14], [42]–[44] typically attempt to solve (1) or its variants in an alternating manner by performing a sparse coding step (updating Z) followed by a dictionary update step (updating D). Methods such as K-SVD [4] also partially update the coefficients in Z in the dictionary update step, while some recent methods update the variables jointly and iteratively [45]. Algorithms for (1) typically repeatedly update Z (involves an NP-hard sparse coding problem) and tend to be computationally expensive.

The DINO-KAT learning problem [13] is an alternative dictionary learning framework that imposes a low-rank constraint on reshaped dictionary atoms (columns). The corresponding problem formulation is

$$\min_{D,Z} \|P - DZ\|_F^2 + \lambda^2 \|Z\|_0$$

$$\text{s.t. } \mathbf{rank}(\mathcal{R}(d_i)) \leq r, \|d_i\|_2 = 1, \|z_l\|_\infty \leq L, \forall i, l, \quad (2)$$

where the ℓ_0 "norm" counts the total number of nonzeros in Z. The operator $\mathcal{R}(\cdot)$ reshapes dictionary atoms $d_i \in \mathbb{C}^n$ into matrices of size $n_1 \times n_2$ for some n_1 and n_2 such that $n = n_1 n_2$, and $r > 0$ is the maximum allowed rank for each reshaped atom. The dimensions of the reshaped atoms can be chosen on an application-specific basis. For example, when learning D from 2D image patches, the reshaped atoms can have the dimensions of the 2D patch. In the case where spatiotemporal (3D) patches are vectorized and extracted from dynamic data,

the atoms could be reshaped into space-time (2D) matrices. Spatiotemporal patches of videos often have high temporal correlation, so they may be well represented by a dictionary with low-rank space-time (reshaped) atoms [13]. The parameter $\lambda > 0$ in (2) controls the overall sparsity of Z. Penalizing the overall or aggregate sparsity of Z enables variable sparsity levels across training signals (a more flexible model than (1)). The ℓ_∞ constraints for $L > 0$ prevent pathologies (e.g., unbounded algorithm iterates) due to the non-coercive objective [46]. In practice, we set L very large, and the constraint is typically inactive. We proposed an efficient algorithm in [13] for Problem (2) that requires less computation than algorithms such as K-SVD.

Another alternative to the DL problems in (1) and (2) involves replacing the constraints in (2) with the unitary constraint $D^H D = I$, where I is the $n \times n$ identity matrix. Learned unitary operators work well in image denoising and more general reconstruction problems [47], [48]. Moreover, algorithms for learning unitary dictionaries tend to be computationally cheap [47]. Our OnAIR framework exploits some of the aforementioned dictionary structures.

III. PROBLEM FORMULATIONS

This section formulates data-driven online image reconstruction. First, we propose an online image reconstruction framework based on an adaptive dictionary regularizer as in (2). Let $\{f^t \in \mathbb{C}^{N_x \times N_y}\}$ denote the sequence of dynamic image frames to be reconstructed. We assume that noisy, undersampled linear measurements of these frames are observed. We process the streaming measurements in minibatches, with each minibatch containing measurements of $\tilde{M} \geq 1$ consecutive frames. Let x^t denote the vectorized version of the 3D tensor obtained by (temporally) concatenating $\tilde{M}$ consecutive frames of the unknown dynamic images. In practice, we construct the sequence $\{x^t\}$ using a sliding window (over time) strategy, which may involve overlapping or non-overlapping minibatches. We model the spatiotemporal (3D) patches of each x^t as sparse with respect to a latent dictionary D. Under this model, we propose to solve the following online dictionary learning-driven image reconstruction problem for each time $t = 1, 2, 3, \ldots$

(P1) $\{\hat{x}^t, \hat{D}^t, \hat{Z}^t\} = \displaystyle\arg\min_{x^t, D, Z^t} \frac{1}{K_t} \sum_{j=1}^{t} \rho^{t-j} \|y^j - A^j x^j\|_2^2$

$$+ \frac{\lambda_S}{K_t} \sum_{j=1}^{t} \rho^{t-j} \left(\sum_{l=1}^{M} \|P_l x^j - D z_l^j\|_2^2 + \lambda_Z^2 \|Z^j\|_0 \right)$$

$$\text{s.t. } \|z_l^t\|_\infty \leq L, \mathbf{rank}(\mathcal{R}(d_i)) \leq r, \|d_i\|_2 = 1, \forall i, l.$$

Here j indexes time, and y^t denotes the (typically undersampled) measurements that are related to the underlying frames x^t (that we would like to reconstruct) through the linear sensing operator A^t. For example, in video inpainting, A^t samples a subset of pixels in x^t, or in dynamic MRI, A^t corresponds to an undersampled Fourier encoding. Typically—and, in particular, in this work—the subsampling occurs per frame using a different random sampling pattern for each frame. For example, in dMRI, measurements are pooled in time to form sets of k-space data (the

corresponding underlying object is often written in the form of a Casorati matrix [25], whose rows represent voxels and columns denote temporal frames). The operator P_l is a patch extraction matrix that extracts the lth $n_x \times n_y \times n_t$ spatiotemporal patch from x^t as a vector. A total of M (possibly) overlapping 3D patches are assumed. Matrix $D \in \mathbb{C}^{n \times m}$ with $n = n_x n_y n_t$ is the synthesis dictionary to be learned and $z_l^t \in \mathbb{C}^m$ is the unknown sparse code for the lth patch of x^t, with $P_l x^t \approx D z_l^t$. Matrix Z^t has z_l^t as its columns, and the ℓ_0 "norm" in (P1) penalizes the aggregate sparsity (i.e., the total number of nonzeros) of Z^t. The weights $\lambda_S, \lambda_Z \geq 0$ are regularization parameters that control the relative adaptive dictionary regularization and sparsity of Z, respectively, in the model. The parameter $0 < \rho \leq 1$ is an exponential forgetting factor that controls the influence of old (previous values of t) data in (P1), and $K_t = \sum_{j=1}^{t} \rho^{t-j}$ is a normalization constant for the objective.

Problem (P1) has a form often used for online optimization [14], [49]. The objective is a weighted average of the function $h(y^t, D, x^t, Z^t) = \|y^t - A^t x^t\|_2^2 + \lambda_S(\sum_{l=1}^{M} \|P_l x^t - D z_l^t\|_2^2 + \lambda_Z^2 \|Z^t\|_0)$, with the constraints defining the feasible sets for D and the sparse codes. K_t serves as a normalization of the objective so that the cost can be viewed as a weighted average of the instantaneous cost $h(y^t, D, x^t, Z^t)$.[2, 3] When $\rho = 1$, this normalization also prevents the cost from becoming unbounded as $t \to \infty$. In practice, this is not an issue when processing finite-size time-series data or with $\rho < 1$. Hence, for simplicity, we may omit this normalization constant.

Problem (P1) jointly estimates the adaptive dictionary model for the patches of x^t together with the underlying image frames. Note that, for each time index t, we solve (P1) only for the latest group of frames x^t and the latest sparse coefficients Z^t, while the previous images and sparse coefficients are set to their estimates from previous minibatches (i.e., $x^j = \hat{x}^j$ and $Z^j = \hat{Z}^j$ for $j < t$). However, the dictionary D is adapted to all the spatiotemporal patches observed up to time t. We emphasize the global dependence of D on all previous data by using the optimization variable D within the objective rather than a time-indexed variable as done for the variables x^t and Z^t. Assuming $\rho = 1$, the objective in (P1) with respect to D acts as a surrogate (upper bound) for the usual empirical (batch) risk function [14], [49] that uses the optimal reconstructions and sparse codes (i.e., those that minimize the cost) for all the data. The exponential factor ρ^{t-j} diminishes the influence of "old" data on the dictionary adaptation process. When the dynamic object or scene changes slowly over time, a large ρ (close to 1) is preferable so that past information has more influence and vice versa.

As written in (P1), the dictionary D is updated based on patches from all previous times; however, the proposed

algorithm does not need to store this information during optimization. Indeed, our algorithm in Section IV computes only a few constant-sized matrices that contain the necessary cumulative (over time) information to solve (P1).

When minibatches x^t and x^{t+1} do not overlap (i.e., no common frames), each frame f^t is reconstructed exactly once in its corresponding window in (P1). However, it is often beneficial to construct x^t using an overlapping sliding window strategy [17], in which case a frame f^t may be reconstructed in multiple windows (minibatches of frames). In this case, we independently produce estimates $\hat{x}^t$ for each time index as indicated in (P1), and then we produce a final estimate of the underlying frame f^t by computing a weighted average of the reconstructions of that frame from each window in which it appeared. We found empirically that an exponentially ρ-weighted average (similar to that in (P1)) performed better than alternatives such as an unweighted average of the estimates from each window or using only the most recent reconstruction from the latest window.

In (P1), we imposed a low-rank constraint on the dictionary atoms. As an alternative, we consider constraining the dictionary to be a unitary matrix. The online optimization problem in this case is as follows:

$$(\text{P2}) \; \{\hat{x}^t, \hat{D}^t, \hat{Z}^t\} = \underset{x^t, D, Z^t}{\arg\min} \frac{1}{K_t} \sum_{j=1}^{t} \rho^{t-j} \|y^j - A^j x^j\|_2^2$$

$$+ \frac{\lambda_S}{K_t} \sum_{j=1}^{t} \rho^{t-j} \left(\sum_{l=1}^{M} \|P_l x^j - D z_l^j\|_2^2 + \lambda_Z^2 \|Z^j\|_0 \right)$$

$$\text{s.t. } D^H D = I,$$

where all terms in (P2) are defined as in (P1). Note that (P2) does not require the ℓ_∞-norm constraints on the sparse coefficients Z^t because the unitary constraint on the dictionary precludes the possibility of repeated dictionary atoms, which was the motivation for including these constraints in (P1) [46].

IV. ALGORITHMS AND PROPERTIES

This section presents the algorithms for Problems (P1) and (P2) and their properties. We propose an alternating minimization-type scheme for (P1) and (P2) and exploit the online nature of the optimization to minimize the costs efficiently. At each time index t, we alternate a few times between updating (D, Z^t) while holding x^t fixed (the *dictionary learning step*) and then updating x^t with (D, Z^t) held fixed (the *image update step*). For each t, we use a warm start for (initializing) the alternating approach. We initialize the dictionary D with the most recent dictionary ($\hat{D}^{t-1}$). Frames of x^t that were estimated in the previous (temporal) windows are initialized with the most recent ρ-weighted reconstructions, and new frames are initialized using simple approaches (e.g., interpolation in the case of inpainting). Initializing the sparse coefficients Z^t with the codes estimated in the preceding window ($\hat{Z}^{t-1}$) worked well in practice. All updates are performed efficiently and with modest memory usage as will be shown next. Fig. 1 provides a graphical flowchart depicting our proposed online scheme. The

---

[2] We include the normalization constant K_t in (P1) to parallel the cost functions in online optimization works in the literature [14], [49].

[3] The minimization in (P1) (and later in (P2)) is only with respect to the variables x^t, D, and Z^t for time t, so one could omit the summation terms involving x^j and Z^j for $j < t$ if desired. However, we retain these terms to emphasize the connection to existing methods in the online optimization literature [14], [49].
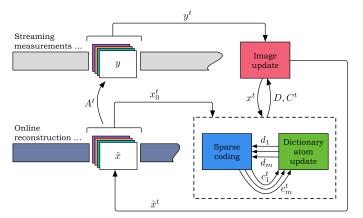
Fig. 1. Flowchart of the proposed online adaptive dictionary learning-driven image reconstruction scheme at time t. The input is a vector y^t containing the streaming measurements for the current minibatch of frames, $\hat{x}^t$ denotes the corresponding reconstructed minibatch, and x^t denotes intermediate reconstructions in the alternating scheme. In the *dictionary learning step* (D, C^t) are updated with x^t held fixed, by performing block coordinate descent over the columns of C^t (*sparse coding*) and the columns of D (*dictionary atom update*). Then, the frames x^t are updated (the *image update step*) with (D, C^t) held fixed. This process is repeated a few times, and the final frame estimates $\hat{x}^t$ are integrated into the streaming reconstruction, $\hat{x}$. x_0^t denotes the initialization (warm start).

next subsections present the alternating scheme for each time t in more detail.

A. Dictionary Learning Step for (P1)

Let $C^t \triangleq (Z^t)^H$. Minimizing (P1) with respect to (D, C^t) yields the following optimization problem:

$$\min_{D, C^t} \sum_{j=1}^{t} \rho^{t-j} \|P^j - D(C^j)^H\|_F^2 + \lambda_Z^2 \|C^t\|_0$$
$$\text{s.t. } \|c_i^t\|_\infty \leq L, \ \mathbf{rank}(\mathcal{R}(d_i)) \leq r, \ \|d_i\|_2 = 1 \ \forall i, \quad (3)$$

where $P^j \in \mathbb{C}^{n \times M}$ is the matrix whose columns contain the patches $P_l x^j$ for $1 \leq l \leq M$, and c_i^t is the ith column of C^t. We use a block coordinate descent approach [46] (with few iterations) to update the sparse coefficients c_i^t and atoms d_i (columns of D) in (3) sequentially. For each $1 \leq i \leq m$, we first minimize (3) with respect to c_i^t keeping the other variables fixed (*sparse coding step*), and then we update d_i keeping the other variables fixed (*dictionary atom update step*). These updates are performed in an efficient online manner as described in the following subsections.

Note that to theoretically converge to a solution of Problem (3), the block coordinate descent updates described here would have to be iterated many times. Fortunately, exhaustive subproblem updates are not necessary in practice for online sliding window optimization problems [14], [49], because warm starts of variables from the previous window(s) can be used as high quality initializations for the updates in the current time window. As such, a few iterations suffice to further update the variables based on new data. In the proposed OnAIR algorithm, the dictionary is continuously updated over many iterations to adapt to the time-series of measurements. If the streaming data all agree with a common dictionary model, it is reasonable to

expect that the online dictionary updates presented below will converge over time.

1) Sparse Coding Step: Minimizing (3) with respect to c_i^t leads to the following subproblem:

$$\min_{c_i^t \in \mathbb{C}^M} \|E_i^t - d_i(c_i^t)^H\|_F^2 + \lambda_Z^2 \|c_i^t\|_0$$
$$\text{s.t. } \|c_i^t\|_\infty \leq L, \quad (4)$$

where the matrix

$$E_i^t := P^t - \sum_{k \neq i} d_k(c_k^t)^H \quad (5)$$

is defined based on the most recent estimates of the other atoms and sparse coefficients. The solution to (4), assuming $L > \lambda_Z$, is given by [46]

$$\hat{c}_i^t = \min\left(|H_{\lambda_Z}((E_i^t)^H d_i)|, \ L1_M\right) \odot e^{j\angle(E_i^t)^H d_i}, \quad (6)$$

where $H_{\lambda_Z}(\cdot)$ is the elementwise hard thresholding operator that sets entries with (complex) magnitude less than λ_Z to zero and leaves other entries unaffected, 1_M is a length-M vector of ones, $\odot$ and $\min(\cdot, \cdot)$ denote elementwise multiplication and elementwise minimum respectively, and $e^{j\angle\cdot}$ is computed elementwise, with $\angle$ denoting the phase. We do not construct E_i^t in (6) explicitly; rather we efficiently compute the matrix-vector product $(E_i^t)^H d_i = (P^t)^H d_i - C^t D^H d_i + c_i^t$ based on the most recent estimates of each quantity using sparse matrix-vector operations [46].

2) Dictionary Atom Update Step: Here, we minimize (3) with respect to d_i. This update uses past information via the forgetting factor ρ. Let $\tilde{P}^j := \sqrt{\rho^{t-j}} P^j$ and $\tilde{C}^j := \sqrt{\rho^{t-j}} C^j$ denote the ρ-weighted patches and sparse coefficients, respectively, and let $\tilde{P}^{1:t}$ and $\tilde{C}^{1:t}$ denote the matrices formed by stacking the $\tilde{P}^j$'s horizontally and $\tilde{C}^j$'s vertically, respectively, for times 1 to t. Finally, define $\tilde{E}_i^{1:t} := \tilde{P}^{1:t} - \sum_{k \neq i} d_k(\tilde{c}_k^{1:t})^H$ using the most recent estimates of all variables, with $\tilde{c}_k^{1:t}$ denoting the kth column of $\tilde{C}^{1:t}$. Using this notation, the minimization of (3) with respect to d_i becomes

$$\min_{d_i \in \mathbb{C}^n} \|\tilde{E}_i^{1:t} - d_i(\tilde{c}_i^{1:t})^H\|_F^2$$
$$\text{s.t. } \mathbf{rank}(\mathcal{R}(d_i)) \leq r, \ \|d_i\|_2 = 1. \quad (7)$$

Let $U_r \Sigma_r V_r^H$ be the rank-r truncated singular value decomposition (SVD) of the matrix $\mathcal{R}(\tilde{E}_i^{1:t} \tilde{c}_i^{1:t})$ that is obtained by computing the r leading singular vectors and singular values of the full SVD $\mathcal{R}(\tilde{E}_i^{1:t} \tilde{c}_i^{1:t}) := U\Sigma V^H$. Then a solution to (7) is given by [13]

$$\mathcal{R}(\hat{d}_i) = \begin{cases} \frac{U_r \Sigma_r V_r^H}{\|\Sigma_r\|_F}, & \text{if } \tilde{c}_i^{1:t} \neq 0 \\ W, & \text{if } \tilde{c}_i^{1:t} = 0, \end{cases} \quad (8)$$

where W is any matrix of appropriate dimension with rank at most r such that $\|W\|_F = 1$. In our experiments, we set W to be the reshaped first column of the $n \times n$ identity matrix, which worked well.

The main computation in (8) is computing $\tilde{E}_i^{1:t} \tilde{c}_i^{1:t}$, since the SVD of the small $n_y n_x \times n_t$ (i.e., space-time matrix with $n_t < n_y n_x$ typically) matrix $\mathcal{R}(\tilde{E}_i^{1:t} \tilde{c}_i^{1:t})$ has negligible computational cost. In principle, the matrix-vector multiplication

$\tilde{E}_i^{1:t}\tilde{c}_i^{1:t}$ depends on all past information processed by the streaming algorithm; however, it can be recursively computed using constant time and memory. Indeed, observe that

$$
\begin{aligned}
\tilde{E}_i^{1:t}\tilde{c}_i^{1:t} &= \sum_{j=1}^{t}\tilde{E}_i^j\tilde{c}_i^j = \sum_{j=1}^{t}\rho^{t-j}E_i^jc_i^j \\
&= \sum_{j=1}^{t}\rho^{t-j}\left(P^j - D(C^j)^H + d_i(c_i^j)^H\right)c_i^j \\
&= \underbrace{\left[\sum_{j=1}^{t}\rho^{t-j}P^jc_i^j\right]}_{=:q_i^t} - D\underbrace{\left[\sum_{j=1}^{t}\rho^{t-j}(C^j)^Hc_i^j\right]}_{=:g_i^t} \\
&\quad + d_i\underbrace{\left[\sum_{j=1}^{t}\rho^{t-j}\|c_i^j\|^2\right]}_{=[g_i^t]_i},
\end{aligned} \tag{9}
$$

where $[z]_i$ denotes the ith element of a vector z. The vectors q_i^t and g_i^t depend on all previous data, but they can be recursively computed over time as

$$
\begin{aligned}
q_i^t &= \rho q_i^{t-1} + P^tc_i^t, \\
g_i^t &= \rho g_i^{t-1} + (C^t)^Hc_i^t,
\end{aligned} \tag{10}
$$

where for each column index i, the matrix C^t is understood to contain the latest versions of the sparse codes already updated (sequentially) during the dictionary learning step. Using these recursive formulae, the product $\tilde{E}_i^{1:t}\tilde{c}_i^{1:t}$ can be readily computed each time in our algorithm. Thus, the update in (8) can be performed in a fully online manner (i.e., without storing all the past data).

Alternatively, we can collect the vectors q_i^{t-1} and g_i^{t-1} as columns of matrices $Q^{t-1} \in \mathbb{C}^{n\times m}$ and $G^{t-1} \in \mathbb{C}^{m\times m}$, and perform the following recursive updates once at the end of of the overall algorithm for (P1) (i.e., once per minibatch of frames):

$$
\begin{aligned}
Q^t &= \rho Q^{t-1} + P^tC^t \\
G^t &= \rho G^{t-1} + (C^t)^HC^t.
\end{aligned} \tag{11}
$$

Here C^t denotes the final sparse codes estimate from the algorithm for (P1). In this case, when performing the inner update (9), the contributions of the two terms on the right hand side of (10) are incorporated separately. The matrices $Q^t \in \mathbb{C}^{n\times m}$ and $G^t \in \mathbb{C}^{m\times m}$ are small, constant-sized matrices whose dimensions are independent of the time index t and the dimensions of the frame sequence, so they are stored for efficient use in the next minibatch. Moreover, the matrix C^t is sparse, so all the matrix-matrix multiplications in (11) (or the matrix-vector multiplications in (10)) are computed using efficient sparse operations.

B. The Unitary Dictionary Variation

In the case of (P2), unlike for (P1), we do not perform block coordinate descent over the columns of D and $(Z^t)^H$. Instead we minimize (P2) with respect to each of the matrices Z^t and D

directly and exploit simple closed-form solutions for the matrix-valued updates. The following subsections describe the solutions to the Z^t and D update subproblems.

1) Sparse Coding Step: Since D is a unitary matrix, minimizing (P2) with respect to Z^t yields the following subproblem:

$$
\min_{Z^t\in\mathbb{C}^{n\times M}} \|D^HP^t - Z^t\|_F^2 + \lambda_Z^2\|Z\|_0, \tag{12}
$$

where P^t is again the matrix whose lth column is P_lx^t. The solution to (12) is given by the elementwise hard-thresholding (at threshold λ_Z) operation

$$
\hat{Z}^t = H_{\lambda_Z}(D^HP^t). \tag{13}
$$

2) Dictionary Update Step: Minimizing (P2) with respect to D results in the following optimization:

$$
\min_{D\in\mathbb{C}^{n\times n}} \sum_{j=1}^{t}\rho^{t-j}\|P^j - D(C^j)^H\|_F^2 \\
\text{s.t. } D^HD = I, \tag{14}
$$

where $C^j \triangleq (Z^j)^H$ is used for notational convenience. Using the definitions of the matrices $\tilde{P}^{1:t}$ and $\tilde{C}^{1:t}$ from the dictionary atom updates in Section IV-A2, we can equivalently write (14) as

$$
\min_{D\in\mathbb{C}^{n\times n}} \|\tilde{P}^{1:t} - D(\tilde{C}^{1:t})^H\|_F^2 \text{ s.t. } D^HD = I. \tag{15}
$$

Problem (15) is a well-known orthogonal Procrustes problem [50]. The solution is given by

$$
\hat{D} = UV^H, \tag{16}
$$

where $U\Sigma V^H$ is a full SVD of $\tilde{P}^{1:t}\tilde{C}^{1:t}$. Similarly as in (11), let $Q^t = \tilde{P}^{1:t}\tilde{C}^{1:t}$. The matrix $Q^t \in \mathbb{C}^{n\times n}$ can be recursively updated over time according to (11) (or (10)), so the dictionary update step can be performed efficiently and fully online.

C. Image Update Step

Minimizing (P1) or (P2) with respect to the minibatch x^t yields the following quadratic sub-problem:

$$
\min_{x^t\in\mathbb{C}^{N_xN_y\bar{M}}} \|A^tx^t - y^t\|_2^2 + \lambda_S\sum_{l=1}^{M}\|P_lx^t - Dz_l^t\|_2^2. \tag{17}
$$

Problem (17) is a least squares problem with normal equation

$$
\left((A^t)^HA^t + \lambda_S\sum_{l=1}^{M}P_l^TP_l\right)x^t = (A^t)^Hy^t + \lambda_S\sum_{l=1}^{M}P_l^TDz_l^t. \tag{18}
$$

In applications such as video denoising or inpainting, the matrix pre-multiplying x^t in (18) is a diagonal matrix that can be efficiently pre-computed and inverted. More generally, in inverse problems where the matrix pre-multiplying x^t (i.e., $(A^t)^HA^t$ in particular) in (18) is not diagonal or readily diagonalizable (e.g., in dynamic MRI with multiple coils), we minimize (17) by applying an iterative optimization method. One can use any classical algorithm in this case, such as the conjugate gradient (CG) method. For the experiments shown in Section V, we used a few iterations (indexed by i) of the simple proximal gradient

---

OnAIR Algorithms for (P1) and (P2)

Inputs : measurement sequence $\{y^t\}$, weights λ_S and λ_Z, rank r, upper bound L, forgetting factor ρ, number of dictionary learning iterations $\hat{K}$, number of proximal gradient iterations $\tilde{K}$, and the number of outer iterations per minibatch K.

Outputs : reconstructed dynamic image sequence $\{\hat{f}^t\}$.

Initialization: Initial (first) estimates $\{f_0^t\}$ of new incoming (measured) frames, initial $\hat{D}^0$ and $\hat{Z}^0$ at $t = 0$, $Q^0 = 0^{n \times m}$ and $G^0 = 0^{n \times m}$.

For $t = 1, 2, 3, \ldots,$ **do**

Warm Start: Set $C_0^t = (\hat{Z}^{t-1})^H$, $D_0^t = \hat{D}^{t-1}$, and x_0^t is set as follows: frames estimated in previous minibatches are set to a ρ-weighted average of those estimates and new frames are set as in $\{g_0^t\}$.

For $k = 1 : K$ **repeat**

1. **Dictionary Learning Step:** For (P1), update (D_k^t, C_k^t) by sequentially updating the $1 \le i \le m$ coefficient and dictionary columns using (6) and (8) (that uses (9) and (10)) respectively, with $\hat{K}$ iterations and initial $\left(D_{k-1}^t, C_{k-1}^t\right)$. For (P2), update (D_k^t, C_k^t) using $\hat{K}$ alternations between performing (13) and (16) with initial $\left(D_{k-1}^t, C_{k-1}^t\right)$.

2. **Image Update Step:** Update x_k^t using (18) or with $\tilde{K}$ iterations of an optimization scheme (e.g., CG or (19)), with initialization x_{k-1}^t.

End

Recursive Updates: Update Q^t and G^t using (11).

Output Updates: Set $\hat{Z}^t = (C_K^t)^H$, $\hat{D}^t = D_K^t$ and $\hat{x}^t = x_K^t$. For frames not occurring in future windows, $\hat{f}^t$ is set to a ρ-weighted average of estimates over minibatches.

End

Fig. 2. The OnAIR algorithms for Problems (P1) and (P2), respectively. Superscript t denotes the time or minibatch index.

method [51], [52] with updates of the form

$$x^{t,i+1} = \mathbf{prox}_{\tau_i \theta} \left(x^{t,i} - \tau_i (A^t)^H (A^t x^{t,i} - y^t) \right), \quad (19)$$

where $\theta(x) := \lambda_S \sum_{l=1}^{M} \|P_l x - D z_l^t\|_2^2$ and the proximal operator of a function h is $\mathbf{prox}_h(x) := \arg\min_z 0.5\|x - z\|_2^2 + h(z)$. The proximal operation in (19) corresponds to a simple least squares problem that is solved efficiently by inverting a diagonal matrix $I + 2\tau_i \lambda_S \sum_{l=1}^{M} P_l^T P_l$ (arising from the normal equation of the proximal operation), which is pre-computed. A constant step-size $\tau_i = \tau < 2/\|A^t\|_2^2$ suffices for convergence [51]. Moreover, the iterations (19) monotonically decrease the objective in (17) when a constant step size $\tau \le 1/\|A^t\|_2^2$ is used [52].

Fig. 2 summarizes the overall OnAIR algorithms for the online optimization Problems (P1) and (P2). In practice, we typically use more iterations K for reconstructing the first minibatch of frames, to create a good warm start for further efficient online optimization. The initial $\hat{D}^0$ in the algorithm can be set to an analytical dictionary (e.g., based on the DCT or wavelets) and we set the initial $\hat{Z}^0$ to a zero matrix.

D. Cost and Convergence

The computational cost for each time index t of the proposed algorithms for solving the online image reconstruction problems (P1) and (P2) scales as $O(n^2 M)$, where $D \in \mathbb{C}^{n \times m}$ with $m \propto n$ assumed, and M is the number of (overlapping) patches in each temporal window. The cost is dominated by various matrix-vector multiplications. Assuming each window's length $\tilde{M} \ll n$, the memory (storage) requirement for the proposed algorithm scales as $O(nM)$, which is the space required to store the image patches of x^t when performing the updates for (P1) or (P2). Typically the minibatch size $\tilde{M}$ is small (to allow better tracking of temporal dynamics), so the number and maximum temporal width of 3D patches in each window are also small, ensuring modest memory usage for the proposed online methods.

While the overall computational cost and memory requirements for each time index t are similar for the algorithms for (P1) and (P2), the simple matrix-valued forms of the dictionary and sparse code updates for (P2) result in practice, in a several-fold decrease in actual runtimes due to optimizations inherent to matrix-valued computations in modern linear algebra libraries.

We emphasize that the OnAIR framework is *computationally cheap* in the sense that it requires only a small number of efficient updates in order to achieve high quality reconstructions in practice. Our proposed algorithm uses warm start initializations and sliding windows, and, at each step, we perform only a few iterations of descent updates to update the variables based on the new minibatch of data. Since the dictionary is updated continuously over time and the sparse coefficients and frames are also estimated over multiple windows, the warm start procedures produce high quality initializations that enable limited iterations in each time window. Limiting to a few iterations can also prevent over-fitting of the model to the (noise or artifacts in the) new data, which can be a concern especially for smaller values of ρ (less memory). Such algorithmic strategies are also adopted in the online optimization literature [14], [49], where solving each instantaneous problem exhaustively is not the goal (indeed this would be infeasible when processing streaming data); instead, online algorithms efficiently improve/adapt estimates over time, usually with asymptotic convergence guarantees wherein the model converges as $t \to \infty$ to stationary points of the expectation of the instantaneous cost.

The proposed algorithms involve either exact block coordinate descent updates or for example, proximal gradient iterations (with appropriate step size) when the matrix pre-multiplying x^t in (18) is not readily diagonalizable. These updates are guaranteed to monotonically decrease the objectives in (P1) and (P2) for each time index t. Whether the overall iterate sequence produced by the algorithms also converges over time (see e.g., [14]) is an interesting open question that we leave for future work.

V. NUMERICAL EXPERIMENTS

This section presents extensive numerical experiments illustrating the usefulness of the proposed OnAIR methods. We

consider two inverse problem applications in this work: video reconstruction (inpainting) from noisy and subsampled pixels, and dynamic MRI reconstruction from highly undersampled k-t space data. The algorithm for (P1) is dubbed OnAIR-LD when the parameter r is lower than its maximum setting (i.e., low-rank dictionary atoms) and is dubbed OnAIR-FD (i.e., full-rank dictionary atoms) otherwise. The algorithm for (P2) is dubbed OnAIR-UD (unitary dictionary). The following subsections discuss the applications and results.[4] See the supplementary material of this manuscript for additional numerical experiments.

A. Video Inpainting

1) Framework: First, we consider video inpainting or reconstruction from subsampled and potentially noisy pixels. We work with the publicly available videos[5] provided by the authors of the BM4D method [53]. We process the first 150 frames of each video at native resolution. We measure a (uniform) random subset of the pixels in each frame of the videos, and also simulate additive (zero mean) Gaussian noise for the measured pixels in some experiments. The proposed OnAIR methods are then used to reconstruct the videos from their corrupted (noisy and/or subsampled) measurements.

The parameters for the proposed OnAIR methods are chosen as follows. We used a sliding (temporal) window of length $\tilde{M} = 5$ frames with a temporal stride of 1 frame, to reconstruct (maximally overlapping) minibatches of frames. In each window, we extracted $8 \times 8 \times 5$ overlapping spatiotemporal patches with a spatial stride of 2 pixels. We learned a square 320×320 dictionary, and the operator $\mathcal{R}(\cdot)$ reshaped dictionary atoms into 64×5 space-time matrices. We ran the OnAIR algorithms for (P1) and (P2) for $K = 7$ iterations in each temporal window (minibatch), with $\hat{K} = 1$ inner iteration (of block coordinate descent or alternation) for updating (D, Z^t), and used a direct reconstruction as per (18) in the image update step. A forgetting factor $\rho = 0.9$ was observed to work well, and we ran the algorithms for more ($K = 50$) iterations for the first minibatch of data. We initialized dictionary D as the transpose of the 3D discrete cosine transform (DCT) matrix[6] (a sparsifying transform), and the initial sparse codes were zero.

Newly arrived frames were first initialized (i.e., in the first minibatch they occur) using 2D cubic interpolation. We simulated various levels of subsampling of the videos (with and without noise), and we chose $r = 5$ or full-rank atoms in (P1), which outperformed low-rank atoms in the experiments here. The videos in this section have substantial temporal motion, so allowing full-rank atoms enabled the algorithm to better learn the dynamic temporal features of the data. Section V-B demonstrates the usefulness of low-rank atoms in (P1). We tuned the weights

λ_S and λ_Z for (P1) and (P2) to achieve good reconstruction quality at an intermediate undersampling factor (70% missing pixels) for each video, and used the same parameters for other factors.

We measure the performance of our methods using the (3D) peak signal-to-noise ratio (PSNR) metric that is computed as the ratio of the peak pixel intensity in the video to the root mean square reconstruction error relative to the ground truth video. All PSNR values are expressed in decibels (dB).

We compare the performance of the proposed OnAIR-FD ($r = 5$) and OnAIR-UD methods for (P1) and (P2) with that of an identical online algorithm but which uses a fixed DCT dictionary. We refer to this variation henceforward as the Online DCT method. In our experiments, we implemented the Online DCT method as outlined in Fig. 2, but omitting the dictionary atom update step in (8). We also produce reconstructions using a "batch" version of the method for (P1) [13] (with $r = 5$) that processes all video frames jointly; this method is equivalent to the proposed online method for (P1) with $\tilde{M}$ set to the total number of frames in the video. For each comparison method, we used the same patch dimensions, initializations, etc., and tuned the parameters λ_S and λ_Z of each method individually. The one exception is that we used a spatial patch stride of 4 pixels for the Batch Learning method rather than the 2 pixel spatial stride used for the online methods. This change was made because the batch method is memory intensive—it extracts and processes image patches from the entire video concurrently—so it was necessary to process fewer patches to make the computation feasible. We ran the batch learning-based reconstruction for 20 iterations, which worked well. Finally, we also compare the OnAIR reconstructions to baseline reconstructions produced by 2D (frame-by-frame cubic) interpolation and 3D interpolation (using the natural neighbor method in MATLAB).

2) Results: Table I lists the PSNR values (in dB) for the various reconstruction methods for different levels of subsampling (from 50% to 90% missing pixels) of three videos (without added noise) from the BM4D dataset. Table II shows analogous results for the Coastguard video when i.i.d. zero-mean Gaussian noise with 25 dB PSNR was added before sampling the pixels. The proposed OnAIR-FD method typically provides the best PSNRs at higher undersampling factors, and the proposed OnAIR-UD method, which uses a more structured (unitary) dictionary,[7] performs better at lower undersampling factors.

The best PSNRs achieved by the OnAIR methods are typically better (by up to 1.4 dB) than for the Batch Learning scheme. Both OnAIR variations also typically outperform (by up to 2 dB) the Online DCT method, which uses a fixed DCT dictionary—the initial D in our methods.

Figs. 3 and 4 show the original and reconstructed frames for a few representative frames of the Coastguard and Flower Garden videos. Results for multiple methods are shown. Fig. 3 shows that the proposed OnAIR-FD method produces visually more accurate reconstructions of the texture in the waves and also produces fewer artifacts near the boats in the water and the rocks

---

[4]The software to reproduce our results is available at http://web.eecs.umich.edu/~fessler.

[5]The data is available at http://www.cs.tut.fi/~foi/GCF-BM3D.

[6]The 3D DCT matrix used here is obtained as the Kronecker product of three one-dimensional DCTs of dimensions matching the patch dimensions ($8 \times 8 \times 5$). Applying the (separable) 3D DCT transform to a patch corresponds to applying 1D DCTs along the row, column, and temporal dimensions of the patch. Equivalently, the patch is sparse coded in the dictionary D that is the transpose of the 3D DCT.

[7]OnAIR-UD can be interpreted as a unitary sparsifying transform model [6] with D^H denoting the sparsifying transform.

TABLE I
PSNR VALUES IN DECIBELS (dB) FOR VIDEO INPAINTING FOR THREE VIDEOS FROM THE BM4D DATASET AT VARIOUS PERCENTAGES OF MISSING PIXELS. THE
METHODS CONSIDERED ARE THE PROPOSED OnAIR-FD METHOD (I.E., $r = 5$), THE PROPOSED OnAIR-UD METHOD, ONLINE INPAINTING WITH A FIXED DCT
DICTIONARY, THE BATCH DICTIONARY LEARNING-BASED RECONSTRUCTION METHOD WITH $r = 5$, 2-D (FRAME-BY-FRAME CUBIC) INTERPOLATION, AND 3-D
INTERPOLATION. THE BEST PSNR VALUE FOR EACH UNDERSAMPLING LEVEL FOR EACH VIDEO IS IN BOLD

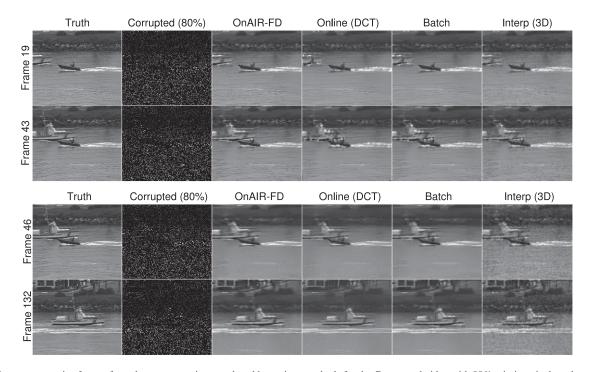| % Missing Pixels	Coastguard					Bus					Flower Garden				
	50%	60%	70%	80%	90%	50%	60%	70%	80%	90%	50%	60%	70%	80%	90%
OnAIR-FD	33.1	**31.4**	**29.6**	**27.3**	22.5	28.7	27.1	**25.5**	**23.7**	**21.5**	**24.4**	**22.8**	**21.0**	**18.8**	15.8
OnAIR-UD	**33.8**	31.3	28.1	24.8	21.9	**29.7**	**27.6**	**25.5**	23.4	21.1	**24.4**	22.1	19.6	17.1	15.6
Online (DCT)	32.7	30.3	27.8	25.3	22.6	28.4	26.7	25.0	23.1	20.8	23.3	21.6	19.9	18.1	16.3
Batch Learning	33.1	31.2	29.1	26.3	22.8	27.8	26.3	24.7	22.9	20.9	23.5	21.8	20.1	18.2	16.1
Interpolation (3D)	29.8	28.5	27.3	25.9	**24.1**	27.3	25.7	24.0	22.1	20.0	20.6	19.6	18.5	17.5	**16.4**
Interpolation (2D)	28.2	26.5	24.9	23.1	21.1	26.0	24.8	23.7	22.5	21.1	20.1	18.8	17.5	16.2	14.8

Fig. 3. Two representative frames from the reconstructions produced by various methods for the Coastguard video with 80% missing pixels and no added noise (top), and 80% missing pixels with 25 dB added Gaussian noise (bottom). Results are shown for the proposed OnAIR-FD ($r = 5$) method, the online method with fixed DCT dictionary, the batch learning-based reconstruction method ($r = 5$), and 3-D interpolation. The true (reference) frames are also shown. Top: OnAIR-FD achieves PSNR (of video) improvements of 2.0 dB, 1.0 dB, and 1.4 dB respectively, over the aforementioned methods. Bottom: OnAIR-FD achieves PSNR improvements of 1.0 dB, 1.0 dB, and 1.5 dB respectively, over the aforementioned methods.
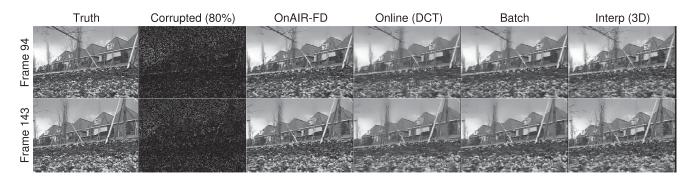

Fig. 4. Two representative frames from the reconstructions produced by various methods for the Flower Garden video with 80% missing pixels. The methods considered are the proposed OnAIR-FD ($r = 5$) method, the online method with fixed DCT dictionary, the batch learning-based reconstruction method, and 3-D interpolation. The true frames are also shown. OnAIR-FD achieves PSNR (of video) improvements of 0.7 dB, 0.6 dB, and 1.3 dB respectively, over the aforementioned competing methods.

TABLE II
PSNR VALUES IN DECIBELS (dB) FOR VIDEO INPAINTING FOR THE COASTGUARD VIDEO CORRUPTED BY GAUSSIAN NOISE WITH 25dB PSNR, AT VARIOUS PERCENTAGES OF MISSING PIXELS. THE FOLLOWING METHODS ARE COMPARED: THE PROPOSED ONAIR-FD ($r = 5$) AND ONAIR-UD SCHEMES, ONLINE INPAINTING WITH A FIXED DCT DICTIONARY, BATCH DICTIONARY LEARNING-BASED RECONSTRUCTION ($r = 5$), 2-D (FRAME-BY-FRAME CUBIC) INTERPOLATION, AND 3-D INTERPOLATION. THE BEST PSNR FOR EACH UNDERSAMPLING IS IN BOLD

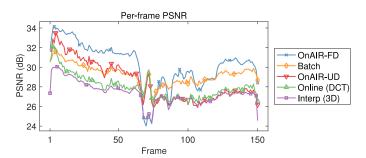| % Missing Pixels	Coastguard (25 dB PSNR)				
	50%	60%	70%	80%	90%
OnAIR-FD	28.6	27.9	**27.2**	**26.1**	**23.9**
OnAIR-UD	**29.4**	**28.6**	26.6	24.9	22.0
Online (DCT)	28.6	27.8	26.7	25.1	22.9
Batch Learning	28.6	27.8	26.6	25.1	22.5
Interpolation (3D)	26.2	25.9	25.4	24.6	23.4
Interpolation (2D)	24.7	24.0	23.0	21.9	20.2

Fig. 5. Per-frame (2D) PSNR values (in dB) for the reconstructions produced by OnAIR-FD ($r = 5$), OnAIR-UD, the online method with fixed DCT dictionary, the batch learning-based method, and 3-D interpolation, for the Coastguard video with 70% missing pixels.

on the shore. Fig. 4 illustrates that the proposed OnAIR method produces a sharper reconstruction with less smoothing artifacts than the online method with a fixed DCT dictionary and the batch learning-based reconstruction method, and it is less noisy than the interpolation-based reconstruction.

3) Properties: Fig. 5 shows the frame-by-frame (2D) PSNRs for the Coastguard video inpainted from 70% missing pixels, using various methods. Clearly the proposed OnAIR-FD method achieves generally higher PSNRs across frames of the video. The decrease in PSNR between frames 70 and 80 in the video is due to the significant motion in the scene that occurs in these frames. This change in performance is typical for dictionary learning-based methods because the sparse codes for the corresponding image patches and dictionary require a few iterations to adapt to the scene. However, as illustrated in Fig. 3, the OnAIR methods may produce visually superior reconstructions in such cases.

To assess the relative efficiency of the OnAIR methods compared to the Batch Learning method, we measured runtimes on the Coastguard video with the same fixed patch sizes, patch strides, and numbers of iterations for each method. The experiments were performed in MATLAB R2017b on a 2016 MacBook Pro with a 2.7 GHz Intel Core i7 processor and 16 GB RAM. With these settings, the OnAIR-FD and OnAIR-UD methods required an average of 4.4 seconds and 1.9 seconds to process each frame, respectively, while the Batch Learning method required an average of 7.9 seconds. Thus the OnAIR-LD and OnAIR-UD methods were 1.8x and 4.2x faster, respectively,

than the Batch Learning method. In addition, the OnAIR methods typically require much fewer iterations (per minibatch) compared to the batch method to achieve their best reconstruction accuracies, so, in practice, when utilizing half as many outer iterations, the OnAIR-FD and OnAIR-UD methods were approximately 3.6x and 8.0x faster, respectively, than the batch scheme.

The decreased memory requirement of the proposed OnAIR method compared to batch learning methods such as from [33] is often crucial in practice. Indeed, even on the modestly sized Coastguard dataset, which contains 150 frames of resolution 144×172, constructing the patch matrix and other associated matrices necessary to run the Batch Learning method with patch strides of $1 \times 1 \times 1$ or $2 \times 2 \times 1$ would have required more than the 16GB of RAM available on our computer (as a result, we chose a $4 \times 4 \times 1$ stride). Moreover, the memory requirement of the Batch Learning method scales linearly with the number of frames (since all frames are processed concurrently), so increasing the length of the frame sequence to, say, 300 frames would again necessitate more RAM than is available on typical machines. Conversely, the OnAIR method has a small, fixed memory footprint that is independent of the number of frames in the dataset (it depends instead on the number of frames *per minibatch*) and thus can process datasets of arbitrary length.

B. Dynamic MRI Reconstruction

1) Framework: Here, we demonstrate the usefulness of the proposed OnAIR methods for reconstructing dynamic MRI data from highly undersampled k-t space measurements. We work with the multi-coil (12-element coil array) cardiac perfusion data [24] and the PINCAT data [12], [54] from prior works. For the cardiac perfusion data, we retrospectively undersampled the k-t space using variable-density random Cartesian undersampling with a different undersampling pattern for each time frame, and for the PINCAT data we used pseudo-radial sampling with a random rotation of radial lines between frames. The undersampling factors tested are shown in Table III. For each dataset, we obtained reconstructions using the proposed OnAIR methods, the online method but with a fixed DCT dictionary, and the batch learning-based reconstruction (based on (P1)) method [13]. We also ran the recent L+S [24] and k-t SLR [55] methods, two sophisticated batch methods for dynamic MRI reconstruction that process all the frames jointly. Finally, we also computed a baseline reconstruction in each experiment by performing zeroth order interpolation across time at non-sampled k-t space locations (by filling such locations with the nearest non-zero entry along time) and then backpropagating the filled k-t space to image space by pre-multiplying with the A^H corresponding to fully sampled data. The first estimates (f_0^t) of newly arrived frames in our OnAIR methods are also computed via such zeroth order interpolation, but using only the estimated (i.e., already once processed and reconstructed) nearest (older) frames.

For the online schemes, we used $8 \times 8 \times 5$ spatiotemporal patches with $\tilde{M} = 5$ frames per temporal window (minibatch) and a (temporal) window stride of 1 frame. We extracted overlapping patches in each minibatch using a spatial stride of 2

TABLE III
NRMSE VALUES AS PERCENTAGES AT SEVERAL UNDERSAMPLING FACTORS FOR THE CARDIAC PERFUSION DATA WITH CARTESIAN SAMPLING (LEFT) AND FOR THE PINCAT DATA WITH PSEUDO-RADIAL SAMPLING (RIGHT). THE METHODS COMPARED ARE THE PROPOSED OnAIR-LD ($r = 1$) METHOD, THE ONLINE SCHEME WITH A FIXED DCT DICTIONARY, THE BATCH LEARNING-BASED RECONSTRUCTION ($r = 1$), THE L+S METHOD, THE K-T SLR METHOD, AND A BASELINE RECONSTRUCTION. THE BEST NRMSE ACHIEVED FOR EACH UNDERSAMPLING AND EACH DATASET IS IN BOLD

Acceleration	Cardiac Perfusion						PINCAT					
	4x	8x	12x	16x	20x	24x	5x	6x	7x	9x	14x	27x
OnAIR-LD	**10.2%**	**12.8%**	**14.8%**	**16.7%**	**18.1%**	**18.0%**	**8.9%**	**9.7%**	**11.0%**	**12.4%**	**15.5%**	21.8%
Online (DCT)	10.8%	13.7%	15.8%	18.2%	20.7%	20.8%	9.5%	10.2%	11.5%	13.2%	16.4%	22.5%
Batch Learning	10.7%	13.7%	15.9%	18.2%	22.0%	23.9%	10.0%	10.7%	11.8%	13.2%	15.9%	**20.9%**
L+S	11.0%	13.8%	16.1%	18.4%	21.5%	22.5%	11.8%	12.9%	14.4%	16.6%	20.0%	25.9%
k-t SLR	11.2%	15.7%	18.4%	21.3%	24.3%	26.5%	9.8%	10.9%	12.4%	14.7%	18.2%	24.2%
Baseline	12.8%	15.9%	18.9%	21.1%	24.5%	28.1%	22.3%	24.7%	27.5%	31.3%	36.6%	44.5%

pixels along each dimension. We learned a square 320×320 dictionary whose atoms when reshaped into 64×5 space-time matrices had rank $r = 1$, which worked well in our experiments. A forgetting factor of $\rho = 0.9$ was observed to work well. We ran the online schemes for $K = 7$ outer iterations per minibatch, with $\hat{K} = 1$ iteration in the dictionary learning step and $\tilde{K} = 10$ proximal gradient steps in the image update step, respectively. Prior to the first outer iteration for each minibatch in (P1), the sparse coefficients were updated using 3 block coordinate descent iterations to allow better adaptation to new patches. We used $K = 50$ outer iterations for the first minibatch to warm start the algorithm and used an initial DCT $\hat{D}^0$, and the initial sparse codes were zero. After one complete pass of the online schemes over all the frames, we performed another pass over the frames, using the reconstructed frames and learned dictionary from the first pass as the first initializations in the second pass. The additional pass gives a slight image quality improvement over the first (a single pass corresponds to a fully online scheme) as will be shown later.

For the batch dictionary learning-based reconstruction method, we used the same patch dimensions, strides, and initializations as for the online methods. We ran the batch method for 50 iterations. For the L+S and k-t SLR methods, we used the publicly available MATLAB implementations from [56] and [57], respectively, and ran each method to convergence. The regularization parameters (weights) for all the methods here were tuned for each dataset by sweeping them over a range of values and selecting values that achieved good reconstruction quality at intermediate k-t space undersampling factors. We measured the dMRI reconstruction quality using the normalized root mean square error (NRMSE) metric expressed as percentage that is computed as

$$\text{NRMSE}(\hat{x}) = \frac{\|\hat{x} - x_{\text{ref}}\|_2}{\|x_{\text{ref}}\|_2} \times 100\%, \qquad (20)$$

where $\hat{x}$ is a candidate reconstruction and x_{ref} is the reference reconstruction (e.g., computed from "fully" sampled data).

2) Results and Comparisons: Table III shows the reconstruction NRMSE values obtained using various methods for the cardiac perfusion and PINCAT datasets at several undersampling factors. The proposed OnAIR-LD method achieves lower NRMSE values in almost every case compared to the L+S and k-t SLR methods, the Online DCT scheme, the baseline reconstruction, and the batch learning-based reconstruction scheme.

In particular, OnAIR-LD achieves a peak improvement of 2.5 dB compared to the Batch Learning method, and it achieves a peak improvement of 1.3 dB compared to the Online DCT method. Unlike the batch schemes (i.e., L+S, k-t SLR, and the batch learning-based reconstruction) that process or learn over all the data jointly, the OnAIR-LD scheme only directly processes data corresponding to a minibatch of 5 frames at any time. Yet OnAIR-LD outperforms the other methods because of its memory (via the forgetting factor) and local temporal adaptivity (or tracking). These results show that the proposed OnAIR methods are well-suited for processing streaming data.

Fig. 6 shows the reconstructions and reconstruction error maps (magnitudes displayed) for some representative frames from the cardiac perfusion and PINCAT datasets at 12x and 7x undersampling, respectively. The error maps indicate that the proposed OnAIR-LD scheme often produces fewer artifacts compared to the existing methods.

Fig. 7 shows the reconstructed $y - t$ profiles for various methods obtained by extracting the same vertical line segment from each reconstructed frame of the PINCAT data and concatenating them. The Online DCT scheme and the batch methods L+S and k-t SLR show line-like or additional smoothing artifacts that are not produced by the proposed OnAIR method, which suggests that the OnAIR method produces reconstructions with greater temporal resolution.

Note that we used full-rank ($r = 5$) atoms in the video inpainting experiments, while in our dynamic MRI reconstruction experiments we chose low-rank ($r = 1$) atoms. Intuitively, low-rank atoms are a better model for the dynamic MRI data because the videos have high temporal correlation and rank-1 atoms are necessarily constant in their temporal dimension. Conversely, the videos used in the inpainting experiments contained significant camera motion and thus dictionary atoms with more temporal variation (i.e., higher rank) enabled more accurate reconstructions.

3) Properties: Table IV investigates the properties of the proposed OnAIR methods in more detail using the cardiac perfusion data. Specifically, it compares the NRMSE values produced by the OnAIR-LD scheme with one or two passes over the data, the OnAIR-UD method, and the online method with a fixed DCT dictionary. In addition, we ran the online method but with a fixed "oracle" dictionary learned from patches of the reference (true) reconstruction by solving the DINO-KAT learning problem (2). The oracle dictionary was computed based on the "fully"
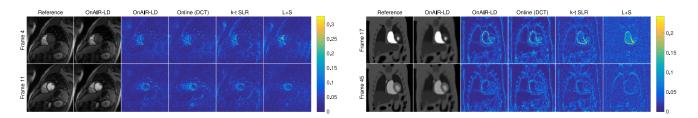
Fig. 6.   Left: reconstructions of the cardiac perfusion data with 12x undersampling (Cartesian sampling). Right: reconstructions of the PINCAT data with 7x undersampling (pseudo-radial sampling). Each panel shows two representative frames from a reference (fully sampled) reconstruction along with the corresponding frames from the proposed OnAIR-LD ($r = 1$) scheme. The right four columns of each panel depict the corresponding reconstruction error magnitudes (w.r.t. reference) for OnAIR-LD, the online method with a fixed DCT dictionary, the k-t SLR method, and the L+S method, respectively. Compared to the competing methods, OnAIR-LD achieves (3-D) NRMSE improvements of 0.6 dB, 1.9 dB, and 0.7 dB, respectively, and 0.4 dB, 1.0 dB, and 2.3 dB, respectively, for the cardiac perfusion and PINCAT data.
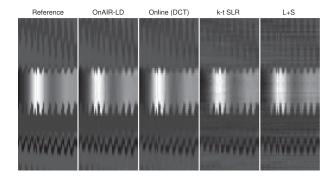

Fig. 7.   Temporal ($y - t$) profiles of a spatial vertical line cross section for the reference PINCAT reconstruction, the proposed OnAIR-LD ($r = 1$) method, the online method with fixed DCT dictionary, the k-t SLR method, and the L+S method for 14x undersampling (pseudo-radial sampling).

TABLE IV
NRMSE VALUES AS PERCENTAGES FOR THE CARDIAC PERFUSION DATA AT SEVERSAL UNDERSAMPLING FACTORS WITH CARTESIAN SAMPLING. THE METHODS COMPARED ARE AN ORACLE ONLINE SCHEME (WHERE THE DICTIONARY IS LEARNED FROM PATCHES OF THE REFERENCE AND FIXED DURING RECONSTRUCTION), THE PROPOSED ONAIR-LD ($r = 1$) METHOD WITH A SINGLE PASS OR TWO PASSES OVER THE FRAMES, THE PROPOSED ONAIR-UD METHOD, AND THE ONLINE SCHEME WITH A FIXED DCT DICTIONARY. THE BEST NRMSE FOR EACH UNDERSAMPLING IS IN BOLD

Acceleration	4x	8x	12x	16x	20x	24x
Online (oracle)	**10.2%**	**12.4%**	**14.4%**	**16.4%**	**17.8%**	**17.9%**
OnAIR-LD (2 passes)	**10.2%**	12.8%	14.8%	16.7%	18.1%	18.0%
OnAIR-LD (1 pass)	**10.2%**	12.9%	14.8%	16.6%	18.3%	18.1%
OnAIR-UD	10.5%	13.6%	15.7%	17.8%	20.4%	20.1%
Online (DCT)	10.8%	13.7%	15.8%	18.2%	20.7%	20.8%

sampled data, so it can be viewed as the "best" dictionary that one could learn from the undersampled dataset. From Table IV, we see that the NRMSE values achieved by the OnAIR-LD scheme with two passes are within 0.0%–0.5% of the oracle NRMSE values, which suggests that the proposed scheme is able to learn dictionaries with good representational (recovery) qualities from highly undersampled data. Moreover, the performance of the OnAIR-LD scheme with a single pass is almost identical to that with two passes, demonstrating the promise of fully online dMRI reconstruction. The OnAIR-LD method outperformed the OnAIR-UD scheme for the cardiac perfusion data indicating that temporal low-rank properties better characterize the dataset than unitary models. The OnAIR schemes achieved NRMSEs that were as much as 13.9% lower than the nonadaptive Online

DCT scheme at higher undersampling rates, which suggests that the learned dictionaries were able to better uncover the hidden structure of the scene.

To assess the relative efficiency of the OnAIR methods compared to the Batch Learning method, we measured runtimes on the PINCAT dataset with the same parameter settings for each method used to generate the results in Table III. With these settings, the OnAIR-LD and OnAIR-UD methods required an average of 8.5 seconds and 2.1 seconds to process each frame, respectively, while the Batch Learning method required an average of 84.0 seconds. Thus the OnAIR-LD and OnAIR-UD methods were 9.9x and 40.0x faster, respectively, than the Batch Learning method. One key reason for these improvements is that 50 outer iterations were required by the batch method while only 7 outer iterations were required by the online methods (after the first minibatch) to achieve the reported reconstruction accuracies; this result suggests that the online dictionary adaptation performed by the OnAIR methods is both computationally efficient and allows the model to better adapt to the underlying structure of the data. The additional speedup of the OnAIR-UD method with respect to OnAIR-LD is attributable to the relative efficiency of the matrix-valued (D, Z) updates of OnAIR-UD compared to the less optimized block coordinate descent iterations over the columns of D and C prescribed by OnAIR-LD.

VI. CONCLUSION

This paper has presented a framework for online estimation of dynamic image sequences by learning dictionaries. Various properties were also studied for the learned dictionary such as a unitary property and low-rank atoms, which offer additional efficiency or robustness to artifacts. The proposed OnAIR algorithms sequentially and efficiently update the images, dictionary, and sparse coefficients of image patches from streaming measurements. Importantly, our algorithms can process arbitrarily long video sequences with low memory usage over time. Our numerical experiments demonstrated that the proposed methods produce accurate reconstructions for video inpainting and dynamic MRI reconstruction. The proposed methods may also be suitable for other inverse problems, including medical imaging applications such as interventional imaging, and other video-processing tasks from computer vision. We hope to investigate other application domains as well as study potential real-time applicability for OnAIR approaches in future work.

References

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

[2] D. Rusanovskyy and K. Egiazarian, "Video denoising algorithm in sliding 3D DCT domain," in *Proc. 7th Int. Conf. Adv. Concepts Intell. Vision Syst.*, 2005, pp. 618–625.

[3] M. Lustig, D. L. Donoho, and J. M. Pauly, "Sparse MRI: The application of compressed sensing for rapid MR imaging," *Magn. Reson. Med.*, vol. 58, no. 6, pp. 1182–1195, 2007.

[4] 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, Nov. 2006.

[5] M. Yaghoobi, T. Blumensath, and M. Davies, "Dictionary learning for sparse approximations with the majorization method," *IEEE Trans. Signal Process.*, vol. 57, no. 6, pp. 2178–2191, Jun. 2009.

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

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

[8] B. Wen, S. Ravishankar, and Y. Bresler, "Learning flipping and rotation invariant sparsifying transforms," in *Proc. IEEE Int. Conf. Image Process.*, 2016, pp. 3857–3861.

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

[10] 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, May 2011.

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

[12] S. G. Lingala and M. Jacob, "Blind compressive sensing dynamic MRI," *IEEE Trans. Med. Imag.*, vol. 32, no. 6, pp. 1132–1145, Jun. 2013.

[13] S. Ravishankar, B. E. Moore, R. R. Nadakuditi, and J. A. Fessler, "Efficient learning of dictionaries with low-rank atoms," in *Proc. IEEE Global Conf. Signal Inf. Process.*, Dec. 2016, pp. 222–226.

[14] J. Mairal, F. Bach, J. Ponce, and G. Sapiro, "Online learning for matrix factorization and sparse coding," *J. Mach. Learn. Res.*, vol. 11, pp. 19–60, 2010.

[15] S. Ravishankar, B. Wen, and Y. Bresler, "Online sparsifying transform learning – Part I: Algorithms," *IEEE J. Sel. Topics Signal Process.*, vol. 9, no. 4, pp. 625–636, Jun. 2015.

[16] S. Ravishankar and Y. Bresler, "Online sparsifying transform learning – Part II: Convergence analysis," *IEEE J. Sel. Topics Signal Process.*, vol. 9, no. 4, pp. 637–646, Jun. 2015.

[17] B. Wen, S. Ravishankar, and Y. Bresler, "Video denoising by online 3D sparsifying transform learning," in *Proc. IEEE Int. Conf. Image Process.*, 2015, pp. 118–122.

[18] B. Wen, S. Ravishankar, and Y. Bresler, "VIDOSAT: High-dimensional sparsifying transform learning for online video denoising," *IEEE Trans. Image Process.*, vol. 28, no. 4, pp. 1691–1704, Apr. 2019.

[19] H. Guo and N. Vaswani, "Video denoising via online sparse and low-rank matrix decomposition," in *Proc. IEEE Statistical Signal Process. Workshop*, Jun. 2016, pp. 1–5.

[20] C. Sutour, C. Deledalle, and J. Aujol, "Adaptive regularization of the NL-means: Application to image and video denoising," *IEEE Trans. Image Process.*, vol. 23, no. 8, pp. 3506–3521, Aug. 2014.

[21] K. Zhang, W. Zuo, Y. Chen, D. Meng, and L. Zhang, "Beyond a gaussian denoiser: Residual learning of deep CNN for image denoising," *IEEE Trans. Image Process.*, vol. 26, no. 7, pp. 3142–3155, Jul. 2017.

[22] M. Lustig, J. M. Santos, D. L. Donoho, and J. M. Pauly, "k-t SPARSE: High frame rate dynamic MRI exploiting spatio-temporal sparsity," in *Proc. ISMRM*, 2006, Paper 2420.

[23] H. Jung, K. Sung, K. S. Nayak, E. Y. Kim, , and J. C. Ye, "k-t FOCUSS: A general compressed sensing framework for high resolution dynamic MRI," *Magn. Reson. Med.*, vol. 61, no. 1, pp. 103–116, 2009.

[24] 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," *Magn. Reson. Med.*, vol. 73, no. 3, pp. 1125–1136, 2015.

[25] Z. P. Liang, "Spatiotemporal imaging with partially separable functions," in *Proc. IEEE Int. Symp. Biomed. Imag.: From Nano Macro*, 2007, pp. 988–991.

[26] B. Dietz, B. G. Fallone, and K. Wachowicz, "Nomenclature for real-time magnetic resonance imaging," *Magn. Reson. Med.*, vol. 81, no. 3, pp. 1483–1484, 2019.

[27] J. P. Haldar and Z. P. Liang, "Spatiotemporal imaging with partially separable functions: A matrix recovery approach," in *Proc. IEEE Int. Symp. Biomed. Imag.: From Nano Macro*, 2010, pp. 716–719.

[28] B. Zhao, J. P. Haldar, C. Brinegar, and Z. P. Liang, "Low rank matrix recovery for real-time cardiac MRI," in *Proc. IEEE Int. Symp. Biomed. Imag.: From Nano Macro*, 2010, pp. 996–999.

[29] H. Pedersen, S. Kozerke, S. Ringgaard, K. Nehrke, and W. Y. Kim, "k-t PCA: Temporally constrained k-t BLAST reconstruction using principal component analysis," *Magn. Reson. Med.*, vol. 62, no. 3, pp. 706–716, 2009.

[30] J. Trzasko and A. Manduca, "Local versus global low-rank promotion in dynamic MRI series reconstruction," in *Proc. ISMRM*, 2011, Paper 4371.

[31] E. J. Candès, X. Li, Y. Ma, and J. Wright, "Robust principal component analysis?" *J. ACM*, vol. 58, no. 3, pp. 11:1–11:37, 2011.

[32] J. Caballero, A. N. Price, D. Rueckert, and J. V. Hajnal, "Dictionary learning and time sparsity for dynamic MR data reconstruction," *IEEE Trans. Med. Imag.*, vol. 33, no. 4, pp. 979–994, Apr. 2014.

[33] S. Ravishankar, B. E. Moore, R. R. Nadakuditi, and J. A. Fessler, "Low-rank and adaptive sparse signal (LASSI) models for highly accelerated dynamic imaging," *IEEE Trans. Med. Imag.*, vol. 36, no. 5, pp. 1116–1128, May 2017.

[34] S. Shalev-Shwartz *et al.*, "Online learning and online convex optimization," *Found. Trends Mach. Learn.*, vol. 4, no. 2, pp. 107–194, 2012.

[35] J. Mairal, F. Bach, J. Ponce, and G. Sapiro, "Online learning for matrix factorization and sparse coding," *J. Mach. Learn. Res.*, vol. 11, pp. 19–60, 2010.

[36] B. E. Moore and S. Ravishankar, "Online data-driven dynamic image restoration using DINO-KAT models," in *Proc. IEEE Int. Conf. Image Process.*, Sep. 2017, pp. 3590–3594.

[37] S. Ravishankar, B. E. Moore, R. R. Nadakuditi, and J. A. Fessler, "Efficient online dictionary adaptation and image reconstruction for dynamic MRI," in *Proc. Asilomar Conf. Signals, Syst., Comput.*, Oct. 2017, pp. 835–839.

[38] R. Gribonval and K. Schnass, "Dictionary identification–sparse matrix-factorization via l_1-minimization," *IEEE Trans. Inf. Theory*, vol. 56, no. 7, pp. 3523–3539, Jul. 2010.

[39] D. Barchiesi and M. D. Plumbley, "Learning incoherent dictionaries for sparse approximation using iterative projections and rotations," *IEEE Trans. Signal Process.*, vol. 61, no. 8, pp. 2055–2065, Apr. 2013.

[40] I. Ramirez, P. Sprechmann, and G. Sapiro, "Classification and clustering via dictionary learning with structured incoherence and shared features," in *Proc. IEEE Int. Conf. Comput. Vision Pattern Recognit.*, 2010, pp. 3501–3508.

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

[42] K. Engan, S. Aase, and J. Hakon-Husoy, "Method of optimal directions for frame design," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 1999, pp. 2443–2446.

[43] K. Skretting and K. Engan, "Recursive least squares dictionary learning algorithm," *IEEE Trans. Signal Process.*, vol. 58, no. 4, pp. 2121–2130, Apr. 2010.

[44] L. N. Smith and M. Elad, "Improving dictionary learning: Multiple dictionary updates and coefficient reuse," *IEEE Signal Process. Lett.*, vol. 20, no. 1, pp. 79–82, Jan. 2013.

[45] A. Rakotomamonjy, "Direct optimization of the dictionary learning problem," *IEEE Trans. Signal Process.*, vol. 61, no. 22, pp. 5495–5506, Nov. 2013.

[46] S. Ravishankar, R. R. Nadakuditi, and J. A. Fessler, "Efficient sum of outer products dictionary learning (SOUP-DIL) and its application to inverse problems," *IEEE Trans. Comput. Imag.*, vol. 3, no. 4, pp. 694–709, Dec. 2017.

[47] S. Ravishankar and Y. Bresler, "Closed-form solutions within sparsifying transform learning," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2013, pp. 5378–5382.

[48] S. Ravishankar and Y. Bresler, 'Data-driven learning of a union of sparsifying transforms model for blind compressed sensing," *IEEE Trans. Comput. Imag.*, vol. 2, no. 3, pp. 294–309, Sep. 2016.

[49] L. Bottou, "Online learning and stochastic approximations," in *On-Line Learning in Neural Networks*. New York, NY, USA: Cambridge Univ. Press, 1998, pp. 9–42.

[50] J. C. Gower, "Generalized procrustes analysis," *Psychometrika*, vol. 40, no. 1, pp. 33–51, 1975.

[51] P. L. Combettes and V. R. Wajs, "Signal recovery by proximal forward-backward splitting," *Multiscale Model. Simul.*, vol. 4, no. 4, pp. 1168–1200, 2005.

[52] N. Parikh and S. Boyd, "Proximal algorithms," *Found. Trends Optim.*, vol. 1, no. 3, pp. 127–239, 2014.

[53] M. Maggioni, V. Katkovnik, K. Egiazarian, and A. Foi, "Nonlocal transform-domain filter for volumetric data denoising and reconstruction," *IEEE Trans. Image Process.*, vol. 22, no. 1, pp. 119–133, Jan. 2013.

[54] B. Sharif and Y. Bresler, "Adaptive real-time cardiac MRI using paradise: Validation by the physiologically improved NCAT phantom," in *Proc. IEEE Int. Symp. Biomed. Imag.: From Nano Macro*, 2007, pp. 1020–1023.

[55] S. G. Lingala, Y. Hu, E. DiBella, and M. Jacob, "Accelerated dynamic MRI exploiting sparsity and low-rank structure: k-t SLR," *IEEE Trans. Med. Imag.*, vol. 30, no. 5, pp. 1042–1054, May 2011.

[56] R. Otazo, "L+S reconstruction matlab code," 2014. Accessed: Mar. 2015. [Online]. Available: http://cai2r.net/resources/software/ls-reconstruction-matlab-code.

[57] S. G. Lingala, "k-t SLR: MATLAB package," 2011. Accessed Mar. 2015. [Online]. Available: http://user.engineering.uiowa.edu/jcb/software/ktslr_matlab/Software.html

Raj Rao Nadakuditi received the Ph.D. degree from the Massachusetts Institute of Technology, Cambridge, MA, USA and the Woods Hole Oceanographic Institution, Woods Hole, MA, USA, in 2007. He is currently an Associate Professor with the Department of Electrical Engineering and Computer Science, University of Michigan, Ann Arbor, MI, USA. His research interests include developing theory for random matrices for applications in signal processing, machine learning, queuing theory, and scattering theory. He was a recipient of an Office of Naval Research Young Investigator Award in 2011, Air Force Office of Scientific Research Young Investigator Award in 2012, Signal Processing Society Young Author Best Paper Award in 2012, and DARPA Young Faculty Award in 2014.

Brian E. Moore received the B.S. degree in electrical engineering from Kansas State University, Manhattan, KS, USA, in 2012, and the M.S. and Ph.D. degrees in electrical engineering from the University of Michigan, Ann Arbor, MI, USA, in 2014 and 2018, respectively.

He is the Co-Founder and CTO of Voxel51, Inc., Ann Arbor, MI. His research interests include efficient algorithms for large-scale machine learning problems with a particular emphasis on computer vision applications.

Saiprasad Ravishankar received the B.Tech. degree in electrical engineering from the Indian Institute of Technology (IIT) Madras, Chennai, India, in 2008, and 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, Champaign, IL, USA. After his Ph.D., he was an Adjunct Lecturer with the Department of Electrical and Computer Engineering, and a Postdoctoral Research Associate with the Coordinated Science Laboratory, University of Illinois. Since August 2015, he has been a Research Fellow with the Department of Electrical Engineering and Computer Science, University of Michigan, Ann Arbor, MI, USA. He was a Postdoc Research Associate with the Theoretical Division, Los Alamos National Laboratory from 2018 to 2019. He is currently an Assistant Professor with the Departments of Computational Mathematics, Science and Engineering, and Biomedical Engineering, Michigan State University, East Lansing, MI. His research interests include signal and image processing, biomedical and computational imaging, data-driven methods, machine learning, signal modeling, inverse problems, data science, compressed sensing, and large-scale data processing. He was a recipient of the IEEE Signal Processing Society Young Author Best Paper Award for 2016. He is currently a member of the IEEE Computational Imaging Technical Committee.

Jeffrey A. Fessler (F'06) received the B.S.E.E. degree from Purdue University, West Lafayette, IN, USA, in 1985, the M.S.E.E. degree from Stanford University, Stanford, CA, USA, in 1986, and the M.S. degree in statistics from Stanford University in 1989, and the Ph.D. degree in electrical engineering from Stanford University in 1990. He is the William L. Root Professor of Electrical Engineering and Computer Science with the University of Michigan, Ann Arbor, MI, USA. From 1985 to 1988, he was a National Science Foundation Graduate Fellow at Stanford. He has worked at the University of Michigan since then. From 1991 to 1992, he was a Department of Energy Alexander Hollaender Post-doctoral Fellow with the Division of Nuclear Medicine. From 1993 to 1995, he was an Assistant Professor in Nuclear Medicine and the Bioengineering Program. He is now a Professor with the Departments of Electrical Engineering and Computer Science, Radiology, and Biomedical Engineering. His research interests include statistical aspects of imaging problems, and he has supervised doctoral research in PET, SPECT, X-Ray CT, MRI, and optical imaging problems. He became a Fellow of the IEEE in 2006, for contributions to the theory and practice of image reconstruction. He was a recipient of the Francois Erbsmann award for his IPMI93 presentation, and the Edward Hoffman Medical Imaging Scientist Award in 2013. He has served as an Associate Editor for the IEEE TRANSACTIONS ON MEDICAL IMAGING, IEEE SIGNAL PROCESSING LETTERS, IEEE TRANSACTIONS ON IMAGE PROCESSING, and IEEE TRANSACTIONS ON COMPUTATIONAL IMAGING. He is currently serving as an Associate Editor for *SIAM Journal on Imaging Science*. He has chaired the IEEE Transactions on Medical Imaging Steering Committee and International Symposium on Biomedical Imaging (ISBI) Steering Committee. He was a Cochair of the SPIE conference on Image Reconstruction and Restoration in 1997, a Technical Program Cochair of the IEEE ISBI in 2002, and a General Chair of ISBI in 2007.

Online Adaptive Image Reconstruction (OnAIR) Using Dictionary Models: Supplementary Material

Brian E. Moore, *Member, IEEE,* Saiprasad Ravishankar, *Member, IEEE,*
Raj Rao Nadakuditi, *Member, IEEE,* and Jeffrey A. Fessler, *Fellow, IEEE*

This supplement contains more exposition to support our manuscript [1]. Specifically, Section VII contains experimental results that provide additional insights into the properties of the proposed OnAIR algorithm.

VII. ADDITIONAL NUMERICAL RESULTS

Here we provide numerical results that extend our investigation from Section V of our manuscript [1].

A. Additional Inpainting Results

Fig. 8 shows the frame-by-frame (2D) PSNRs for the Bus video[1] inpainted from 50% missing pixels using various methods. Clearly the proposed OnAIR-UD scheme achieves consistently higher PSNRs across all frames. The overall trends in PSNR over frames are similar across the methods and are due to motion in the original videos, with more motion generally resulting in lower PSNRs.

Fig. 9 shows a representative example of a (final) learned dictionary produced by the proposed OnAIR-FD method for the Bus video, along with the initial (at $t = 0$) DCT dictionary. The dictionaries each contain 320 atoms, each of which is an $8 \times 8 \times 5$ space-time tensor. We visualize each atom by plotting the first 8×8 $x - y$ slice of the atom and also plotting the $y - t$ profile from a vertical slice through the middle of each atom tensor. The $x - y$ (first slice) images show that the learned dictionary has adapted to both smooth and sharp gradients in the image, and the dynamic (evolved) nature of the $y - t$ profiles shows that the dictionary atoms have adapted to temporal trends in the data.

B. Additional Dynamic MRI Reconstruction Results

Fig. 10 shows that the OnAIR-LD scheme typically achieves better frame-by-frame NRMSE compared to the other dynamic MRI reconstruction methods considered in this work. Note that such higher quality reconstructions are obtained in spite of the fact that the online scheme only processes and stores data corresponding to 5 frames (in x^t) at any time while the

[1]The video was provided by the authors of the BM4D method [2]. The dataset is publicly at http://www.cs.tut.fi/~foi/GCF-BM3D.
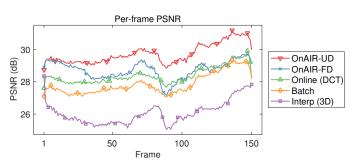

Fig. 8: Per-frame (2D) PSNR values (in dB) for the reconstructions produced by OnAIR-FD ($r = 5$), OnAIR-UD, Online DCT, the Batch Learning method, and 3D interpolation, for the Bus video with 50% missing pixels.
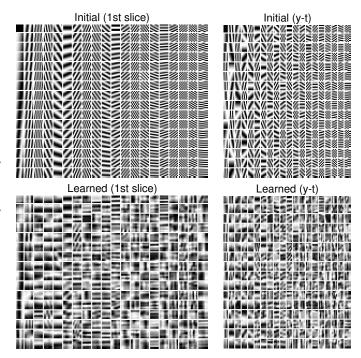

Fig. 9: Dictionaries for the Bus video with 50% missing pixels. Top: the initial DCT dictionary. Bottom: the final learned dictionary produced by OnAIR-FD. Left: the first 8×8 ($x - y$) slice of each $8 \times 8 \times 5$ dictionary atom. Right: the $y - t$ profiles from a vertical cross-section through each $8 \times 8 \times 5$ atom.

L+S, k-t SLR, and batch DINO-KAT methods process all data jointly.

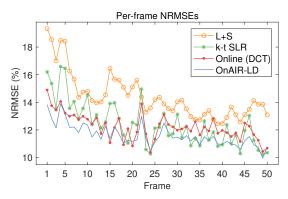Fig. 11 shows an example of a learned dictionary produced

Fig. 10: Per-frame PSNR values for the reconstructions produced by OnAIR-LD ($r = 1$), Online DCT, and the k-t SLR and L+S methods, for the PINCAT data from [3], [4] with 9x undersampling and pseudo-radial sampling.
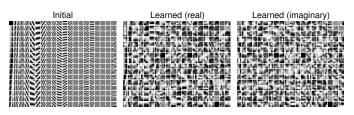

Fig. 11: Dictionaries for the PINCAT data from [3], [4] with 9x undersampling. Left: the atoms of the initial DCT dictionary. Center and right: the real and imaginary parts of the final learned dictionary produced by OnAIR-LD ($r = 1$). The dictionary atoms are $8 \times 8 \times 5$ tensors, and only the first 8×8 ($x - y$) slice of each atom is displayed.

by the proposed OnAIR-LD method on the PINCAT dataset at 9x undersampling, which is compared with the initial DCT dictionary. The dictionaries have 320 atoms, each a $8 \times 8 \times 5$ complex-valued space-time tensor. Since the OnAIR LD dictionary atoms have rank $r = 1$ when reshaped into 64×5 space-time matrices, we directly display the real and imaginary parts of the first 8×8 ($x - y$) slice of each learned atom. The initial DCT is also displayed similarly. The (eventual) learned dictionary has clearly evolved significantly from the initial DCT atoms and has adapted to certain smooth and sharp textures at various orientations in the data.

REFERENCES

[1] B. E. Moore, S. Ravishankar, R. R. Nadakuditi, and J. A. Fessler, "Online adaptive image reconstruction (OnAIR) using dictionary models," *IEEE Transactions on Computational Imaging*, 2019.

[2] M. Maggioni, V. Katkovnik, K. Egiazarian, and A. Foi, "Nonlocal transform-domain filter for volumetric data denoising and reconstruction," *IEEE transactions on image processing*, vol. 22, no. 1, pp. 119–133, 2013.

[3] S. G. Lingala and M. Jacob, "Blind compressive sensing dynamic MRI," *IEEE Transactions on Medical Imaging*, vol. 32, no. 6, pp. 1132–1145, 2013.

[4] B. Sharif and Y. Bresler, "Adaptive real-time cardiac MRI using paradise: Validation by the physiologically improved NCAT phantom," in *IEEE International Symposium on Biomedical Imaging: From Nano to Macro*, 2007, pp. 1020–1023.