Online Dictionary Learning Algorithm with Periodic Updates and its Application to Image Denoising

Ender M. Eksioglu

Electronics and Communications Engineering Department
Istanbul Technical University
Istanbul, Turkey
e-mail: eksioglue@itu.edu.tr
Phone/fax: +90-212-2853623

Abstract

We introduce a coefficient update procedure into existing batch and online dictionary learning algorithms. We first propose an algorithm which is a coefficient updated version of the Method of Optimal Directions (MOD) dictionary learning algorithm (DLA). The MOD algorithm with coefficient updates presents a computationally expensive dictionary learning iteration with high convergence rate. Secondly, we present a periodically coefficient updated version of the online Recursive Least Squares (RLS)-DLA, where the data is used sequentially to gradually improve the learned dictionary. The developed algorithm provides a periodical update improvement over the RLS-DLA, and we call it as the Periodically Updated RLS Estimate (PURE) algorithm for dictionary learning. The performance of the proposed DLAs in synthetic dictionary learning and image denoising settings demonstrates that the coefficient update procedure improves the dictionary learning ability.

Keywords: Dictionary learning, sparse representation, online learning, image denoising

1. Introduction

Sparse signal representation in overcomplete dictionaries has acquired considerable interest [1, 2, 3]. Sparse signal representation constitutes compactly expressing a signal as a linear combination from an overcomplete set of signals or atoms. The number of atoms utilized in the linear combination is much less than the signal dimensionality, hence the sparse designation. The set of all atoms forms the redundant dictionary over which sparse representations are realized. There are a plethora of methods for sparse representation of a signal over a given dictionary [4]. One class of algorithms includes linear programming based optimization methods [5]. Another important class of algorithms contain the greedy methods, e.g., Orthogonal Matching Pursuit (OMP) [6], which present computationally practical solutions to the sparse representation problem.

A subject related to sparse representation is dictionary learning [1, 7, 8, 9], which considers the construction of the dictionary employed for sparse coding of data. Dictionary learning examines the problem of training the atoms of a dictionary suitable for the joint sparse representation of a data set. Dictionary learning algorithms (DLAs) include Maximum Likelihood (M-L) methods [10], Maximum a-posteriori Probability (MAP)-based methods [11], the K-Singular Value Decomposition (K-SVD) algorithm [12], direct optimization based methods such as [13] and the least-squares based Method of Optimal Directions (MOD) [14, 15]. Other recent approaches to the dictionary learning problem include [16, 17].

In general the previously listed methods are batch algorithms, and they process the entire data set as a batch for each iteration. Recently, online

DLAs have been proposed, where the algorithm allows sequential dictionary learning as the data flows in. The online algorithms include the Recursive Least Squares (RLS)-DLA [18], which is derived using an approach similar to the RLS algorithm employed in adaptive filtering. The RLS approach has also been used for sparse adaptive filtering in recent studies [19, 20]. Another online DLA is the Online Dictionary Learning (ODL) algorithm of [21].

In this paper we introduce a new DLA, which is based on the least squares solution for the dictionary estimate as is the case for the MOD algorithm and the RLS-DLA. We first present a variant of the MOD algorithm where the sparse coefficients associated with the previously seen signals are recalculated at every iteration before the dictionary is updated. This variant has much higher computational complexity than the MOD algorithm. We regularize this computationally expensive variant by restricting the recalculation to periodic updates. The resulting algorithm which we call as the PURE algorithm is developed by augmenting the RLS-DLA algorithm with periodic updates of the sparse representations before the dictionary estimate is formed. The PURE algorithm presents performance better than the RLS-DLA, while maintaining the same asymptotic computational complexity as the RLS-DLA and MOD algorithms. Simulations show that the introduced PURE algorithm works well in the synthetic dictionary reconstruction setting and also in image denoising applications. To the best of our knowledge this work presents the first attempt to introduce a periodic coefficient update into the two-step iterative dictionary learning procedure. Dictionary learning for given data sets results in performance improvement in various applications. These applications include but are not limited to image denoising and reconstruction [22, 23, 24] and various classification problems [25]. Devising new and better dictionary learning approaches naturally leads to performance improvements in the aforementioned applications.

In the coming sections, we begin first by giving a review of dictionary learning in general, and the MOD and RLS-DLA algorithms. In Section 3 we introduce the coefficient updated version of the MOD algorithm. In Section 4, we develop a new online dictionary learning algorithm by augmenting the RLS-DLA with periodic coefficient updates. Section 5 details the computational complexity of the novel algorithms when compared to the existing methods. In Section 6 we provide detailed simulations for the novel algorithms. The simulation settings include synthetic dictionary recovery and image denoising.

2. Batch and Online Dictionary Learning Algorithms

The dictionary learning problem may be defined as finding the optimally sparsifying dictionary for a given data set. The dictionary learning problem might be formulated using different optimization objectives over a sparsity regularized cost function for a given data set. [12] suggests the following expression for constructing a sparsifying dictionary.

$$\min_{\mathbf{D},\mathbf{W}}\left\{\sum_{n=1}^{N}\|\boldsymbol{x}_n-\mathbf{D}\boldsymbol{w}_n\|_2^2\right\} \text{ subject to } \forall n, \|\boldsymbol{w}_n\|_0 \leq S \tag{1}$$

or equivalently

$$\min_{\mathbf{D},\mathbf{W}}\left\{\|\mathbf{X}-\mathbf{D}\mathbf{W}\|_F^2\right\} \text{ subject to } \forall n, \|\boldsymbol{w}_n\|_0 \leq S \tag{2}$$

Another similar objective for dictionary learning considered in [12] is

$$\min_{\mathbf{D},\mathbf{W}} \left\{ \sum_{n=1}^{N} \|\boldsymbol{w}_n\|_0 \right\} \text{ subject to } \forall n, \|\mathbf{X} - \mathbf{DW}\|_F^2 \leq \epsilon. \tag{3}$$

$\|\cdot\|_F$ is the Frobenious norm for the matrix argument, and $\|\cdot\|_0$ is the ℓ_0 pseudo-norm for a vector argument. $\mathbf{X} \in \mathbb{R}^{M\times N}$ is the data matrix, which stores all the data vectors for time $n = 1$ through N. $\mathbf{X} = [\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N]$, where N is the total number of observed data vectors, and $\boldsymbol{x}_n \in \mathbb{R}^M$ is the data vector at time n. $\mathbf{D} \in \mathbb{R}^{M\times K}$ is the dictionary matrix with K atoms as columns, that is $\mathbf{D} = [\boldsymbol{d}_1, \ldots, \boldsymbol{d}_K]$. $\boldsymbol{w}_n \in \mathbb{R}^K$ is the sparse representation vector for $\boldsymbol{x}_n$, and $\mathbf{W} \in \mathbb{R}^{K\times N}$ is the sparse representation weight matrix, $\mathbf{W} = [\boldsymbol{w}_1, \ldots, \boldsymbol{w}_N]$. S is the maximum allowed number of nonzero elements for $\boldsymbol{w}_n$. We propose the following formulation for the sparsifying dictionary learning problem.

$$\min_{\mathbf{D},\mathbf{W}} \left\{ \|\mathbf{X} - \mathbf{DW}\|_F^2 + \gamma \sum_{n=1}^{N} \|\boldsymbol{w}_n\|_0 \right\} \tag{4}$$

For appropriate selection of the parameters S, ϵ and γ, we can state that all three formulations (2), (3) and (4) treat the dictionary learning problem in a similar manner, and they all seek the optimal dictionary which results in adequately sparse representations and an acceptable representation error for a given data record $\mathbf{X}$. The main approach utilized by the DLAs in the literature for the solution of the dictionary learning optimization problem is a two-step iterative refinement procedure. In this approach at each step of the iterations either one of $\mathbf{D}$ or $\mathbf{W}$ is held constant, and the optimization is realized over the other matrix. The i^{th} iteration for this two-step iterative refinement approach in batch mode can be summarized as follows.

1) Find sparse $\mathbf{W}^{(i)}$ for constant $\mathbf{D}^{(i-1)}$:

$$\boldsymbol{w}_n^{(i)} = \underset{\boldsymbol{w}}{\text{argmin}} \|\boldsymbol{x}_n - \mathbf{D}^{(i-1)}\boldsymbol{w}\|_2^2 + \gamma\|\boldsymbol{w}\|_0, \text{ for } n = 1, \ldots, N \tag{5}$$

2) Find optimal $\mathbf{D}^{(i)}$ for constant $\mathbf{W}^{(i)}$:

$$\mathbf{D}^{(i)} = \underset{\mathbf{D}}{\text{argmin}} \|\mathbf{X} - \mathbf{D}\mathbf{W}^{(i)}\|_F^2. \tag{6}$$

The first step above is a batch sparse representation problem. Here, the sparse representation or vector selection problem is solved for all the N data vectors separately using the same dictionary matrix $\mathbf{D}^{(i-1)}$. The sparse representation method to apply in this step can be chosen among a multitude of methods from sparse coding literature. The sparse representation methods used by different DLAs include simple gradient descent update [10], FOcal Underdetermined System Solver (FOCUSS) [11], OMP [12] and the Least Angle Regression (LARS) algorithm [21].

The second step is where the DLAs utilizing the two step approach differ from each other. The pioneering work of [10] suggests an ML approach, where gradient descent correction is utilized for the calculation of the updated $\mathbf{D}^{(i)}$.

$$\mathbf{D}^{(i)} = \mathbf{D}^{(i-1)} - \eta \sum_{n=1}^{N} \big(\mathbf{D}^{(i-1)}\boldsymbol{w}_n^{(i)} - \boldsymbol{x}_n\big)\boldsymbol{w}_n^{(i)T} \tag{7}$$

K-SVD [12] uses an SVD based algorithm to update $\mathbf{D}^{(i-1)}$, where the values but not the positions of the non-zero elements of $\mathbf{W}^{(i)}$ can also get updated. The method of optimized directions or the MOD algorithm [14] has also been called as the Iterative Least Squares Dictionary Learning Algorithm or ILS-DLA [15]. MOD has been proposed as a least squares iterative approach for dictionary design from data. The MOD algorithm fits into the iterative

Algorithm 1 MOD algorithm for dictionary learning [15].

Input: Data record of length N, $\mathbf{X} = \mathbf{X}_N = [\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N]$.

```
Initialize the dictionary, possibly as D^(0) = X_K.
for i := 1, 2, ... do                                         ▷ epoch iteration
    for n := 1, 2, ..., N do          ▷ batch iteration for sparse representation
        w_n^(i) = argmin_w ||x_n − D^(i−1) w||_2^2 + γ||w||_0
    end for
    W^(i) = [w_1^(i), ..., w_N^(i)]                  ▷ sparse representation matrix
    D^(i) = X W^(i)† = X W^(i)T [W^(i) W^(i)T]^(−1)        ▷ dictionary update step
end for                                                      ▷ end of iteration
```

relaxation based two-step approach for dictionary design as described above. The MOD algorithm calculates the exact least squares solution for (6).

$$\mathbf{D}^{(i)} = \mathbf{X}\mathbf{W}^{(i)^{\dagger}} = \mathbf{X}\mathbf{W}^{(i)^{T}}\left[\mathbf{W}^{(i)}\mathbf{W}^{(i)^{T}}\right]^{-1} \tag{8}$$

Here, $(\cdot)^{\dagger}$ denotes the Moore-Penrose pseudo-inverse. The outline for the MOD algorithm is presented in Alg.1. Here, by epoch we mean a complete run over the available training set. The common theme for the algorithms introduced in [10, 12, 15, 26] is that in all these algorithms, both steps of the two-step approach are run in batch mode. $\mathbf{D}^{(i)}$ is recalculated only once every epoch after all sparse representation vectors are found using $\mathbf{D}^{(i-1)}$. Recently there have been attempts at online DLA's, where the two-steps are run in an online, streaming data fashion [18, 21]. By the online designation, it is meant that the dictionary is updated continuously in an online fashion for each incoming data vector. In the online mode except the epoch iteration, there is also an iteration over time index. Hence, the two-step approach for the online modality can be formalized as follows.

1) Find optimal $\boldsymbol{w}_n$ for constant $\mathbf{D}_{n-1}$, that is find

$$\boldsymbol{w}_n = \underset{\boldsymbol{w}}{\operatorname{argmin}} \|\boldsymbol{x}_n - \mathbf{D}_{n-1}\boldsymbol{w}\|_2^2, \text{ s.t. } \|\boldsymbol{w}\|_0 \leq S. \tag{9}$$

2) Find optimal $\mathbf{D}_n$ for constant $\mathbf{W}_n$, that is find

$$\mathbf{D}_n = \underset{\mathbf{D}}{\operatorname{argmin}} \|\mathbf{X}_n - \mathbf{D}\mathbf{W}_n\|_F^2. \tag{10}$$

Here, $\mathbf{X}_n = [\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n] \in \mathbb{R}^{M \times n}$ is the partial data matrix. $\mathbf{W}_n = [\boldsymbol{w}_1, \boldsymbol{w}_2, \ldots, \boldsymbol{w}_n] \in \mathbb{R}^{K \times n}$ is the corresponding instantaneous weight matrix. The first step constitutes a sparse representation operation, yet for a single vector $\boldsymbol{x}_n$. The second step is the instantaneous dictionary update, where the innovation gathered from the calculation of the new weight vector $\boldsymbol{w}_n$ is used to revamp $\mathbf{D}_{n-1}$.

Recently an online version of the MOD algorithm denoted as RLS-DLA has been developed [18]. In the online formulation of the RLS-DLA, instantaneous data matrix $\mathbf{X}_n$ and instantaneous sparse representation matrix $\mathbf{W}_n$ replace the matrices in the least squares solution (8). The main algorithmic structure for the online approach of the RLS-DLA algorithm is summarized in Alg.2. In the development of the RLS-DLA algorithm, the relation between the successive matrices $\mathbf{W}_{n-1}$ and $\mathbf{W}_n$ is utilized to conceive an update scheme which calculates $\mathbf{W}_n^{\dagger}$ from $\mathbf{W}_{n-1}^{\dagger}$ without any explicit matrix inversion [18]. The RLS-DLA algorithm prevents matrix inversion at each time step, and hence its computational complexity does not get excessively high. We state that it is possible to design other least-squares solution based DLAs, and we present these novel algorithms in the coming sections.

Algorithm 2 RLS-DLA [18].

Input: Data record of length N, $\mathbf{X} = \mathbf{X}_N = [\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N]$.

1: Initialize the dictionary, possibly as $\mathbf{D}_N^{(0)} = \mathbf{X}_K$.
2: **for** $i := 1, 2, \ldots$ **do** ▷ epoch iteration
3: $\quad \mathbf{D}_0^{(i)} = \mathbf{D}_N^{(i-1)}$, $\mathbf{W}_0 = [\,]$ ▷ initialization
4: $\quad$ **for** $n := 1, 2, \ldots N$ **do** ▷ time iteration
5: $\quad\quad \boldsymbol{w}_n^{(i)} = \underset{\boldsymbol{w}}{\operatorname{argmin}} \|\boldsymbol{x}_n - \mathbf{D}_{n-1}^{(i)} \boldsymbol{w}\|_2^2 + \gamma \|\boldsymbol{w}\|_0$ ▷ sparse representation step
6: $\quad\quad \mathbf{W}_n = [\mathbf{W}_{n-1} | \boldsymbol{w}_n^{(i)}]$ ▷ instantaneous sparse representation matrix
7: $\quad\quad \mathbf{D}_n^{(i)} = \mathbf{X}_n \mathbf{W}_n{}^{\dagger}$ ▷ dictionary update step
8: $\quad$ **end for** ▷ end of time recursion
9: **end for** ▷ end of iteration

3. Least Squares Dictionary Learning Algorithm with Coefficient Update

We present a variation on the MOD algorithm, which presents a time recursion different when compared to the RLS-DLA. The RLS-DLA algorithm as presented in Alg.2, finds only the sparse representation for the current data vector $\boldsymbol{x}_n$ at Step 5. Hence, the current instantaneous weight matrix $\mathbf{W}_n$ is generated by concatenating the previous weight matrix $\mathbf{W}_{n-1}$ with $\boldsymbol{w}_n$, that is $\mathbf{W}_n = [\mathbf{W}_{n-1} |\ \boldsymbol{w}_n]$. We suggest that better convergence in the dictionary update can be achieved if at each time instant also the sparse representations for previous data vectors get updated. Hence, in the proposed variation, the instantaneous weight matrix $\mathbf{W}_n$ is found from scratch by applying batch sparse representation to all the data vectors up to time n. This means that at each time point n, n sparse representation problems should be solved, where

Algorithm 3 MOD with coefficient update (MOD-CU) dictionary learning algorithm.

Input: Data record of length N, $\mathbf{X} = \mathbf{X}_N = [\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N]$.

Initialize the dictionary, possibly as $\mathbf{D}_0 = \mathbf{X}_K$.
for $n := 1, 2, \ldots N$ **do** ▷ time iteration
  for $t := 1, 2, \ldots, n$ **do** ▷ iteration for sparse representation step
    $\boldsymbol{w}_t = \underset{\boldsymbol{w}}{\operatorname{argmin}} \|\boldsymbol{x}_t - \mathbf{D}_{n-1}\boldsymbol{w}\|_2^2 + \gamma\|\boldsymbol{w}\|_0$
  end for
  $\mathbf{W}_n = [\boldsymbol{w}_1, \ldots, \boldsymbol{w}_n]$ ▷ instantaneous sparse representation matrix
  $\mathbf{D}_n = \mathbf{X}_n \mathbf{W}_n{}^{\dagger}$ ▷ dictionary update step
end for ▷ end of time iteration

in Step 5 of Alg.2, only a single sparse coding problem is solved. In this variation the pseudo inverse of the $\mathbf{W}_n$ will no longer be recursively calculable from $\mathbf{W}_{n-1}^{\dagger}$. Hence, a $K \times K$ matrix inversion is required at each time instant. We call this new MOD-based dictionary learning algorithm variant as MOD with coefficient update (MOD-CU), and its steps are detailed in Alg.3. The computational complexity of the proposed algorithm for a single epoch iteration is much larger than the RLS-DLA and the MOD algorithms, as it requires the solution of multiple sparse representation problems at each time point. On the other hand, a single epoch iteration through the data record for MOD-CUalso gives much better dictionary atom recovery results when compared to the RLS-DLA and MOD algorithms. The MOD-CU algorithm as presented in Alg.3 realizes only a single epoch iteration over the whole data set. The resulting dictionary estimate from this process can be used as the initial dictionary for RLS-DLA, MOD or any other DLA. Remaining iterations might be performed using the DLA of choice.

4. RLS-DLA with Periodic Weight Updates: PURE-DLA

At time instant n, the MOD-CU algorithm necessitates solving the sparse representation problem for n data vectors and finding the inverse of a $K \times K$ matrix. The RLS-DLA on the other hand requires solving the sparse representation problem only for the current data vector and requires no explicit matrix inversion, because the matrix inverse is calculated using a rank one update similar to the RLS algorithm. Hence, it is tempting to find an algorithm which maintains the relative performance gain of MOD-CU without severely compromising the computational efficiency of RLS-DLA. We propose a new online dictionary learning algorithm based on the RLS-DLA. In this new algorithm the dictionary estimate of the RLS-DLA is periodically updated at certain time intervals using the dictionary update steps of MOD-CU (Alg.3, steps 3-5). We call this new hybrid algorithm, which maintains the advantages of both RLS-DLA and the coefficient update as the Periodically Updated RLS Estimate (PURE) dictionary learning algorithm. The PURE-DLA algorithm is outlined in Alg.4. In PURE-DLA, the RLS-DLA is realized in its regular form. Steps 5-9 in Alg.4 realize the online update of the dictionary estimate at every time instant as adapted from the RLS-DLA. Step 5 is the instantaneous solution for the current data vector, while steps 6-9 realize the dictionary estimate update using the current sparse representation.

The difference between PURE and RLS-DLA occurs in the periodic dictionary update step which is executed periodically after every P time in-

Algorithm 4 Periodically Updated RLS Estimate dictionary learning algorithm (PURE-DLA) .

Input: Data record of length N, $\mathbf{X} = \mathbf{X}_N = [\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N]$.

1: Initialize the dictionary, possibly as $\mathbf{D}_N^{(0)} = \mathbf{X}_K$.
2: **for** $i := 1, 2, \ldots$ **do** ▷ epoch iteration
3: $\mathbf{D}_0^{(i)} = \mathbf{D}_N^{(i-1)}$, $\mathbf{C}_0 = \mathbf{I}_K$ ▷ initialization
4: **for** $n := 1, 2, \ldots N$ **do** ▷ time iteration
5: $\boldsymbol{w}_n = \underset{\boldsymbol{w}}{\operatorname{argmin}} \|\boldsymbol{x}_n - \mathbf{D}_{n-1}^{(i)} \boldsymbol{w}\|_2^2 + \gamma \|\boldsymbol{w}\|_0$ ▷ RLS update starts with sparse representation step
6: $\boldsymbol{r} = \boldsymbol{x}_n - \mathbf{D}_{n-1}^{(i)} \boldsymbol{w}_n$, $\mathbf{C}^*{}_{n-1} = \lambda^{-1} \mathbf{C}_{n-1}$
7: $\boldsymbol{u} = \mathbf{C}_{n-1}^* \boldsymbol{w}_n$, $\alpha = \dfrac{1}{1 + \boldsymbol{w}_n^T \boldsymbol{u}}$
8: $\mathbf{C}_n = \mathbf{C}_{n-1}^* - \alpha \boldsymbol{u}\boldsymbol{u}^T$
9: $\mathbf{D}_n^{(i)} = \mathbf{D}_{n-1}^{(i)} + \alpha \boldsymbol{r}\boldsymbol{u}^T$ ▷ RLS update finishes
10: **if** $\mod(n, P) = 0$ **then** ▷ periodic update starts
11: **for** $t := 1, 2, \ldots, n$ **do** ▷ sparse representation iteration for periodic update
12: $\boldsymbol{w}_t = \underset{\boldsymbol{w}}{\operatorname{argmin}} \|\boldsymbol{x}_t - \mathbf{D}_n^{(i)} \boldsymbol{w}\|_2^2 + \gamma \|\boldsymbol{w}\|_0$
13: **end for**
14: $\mathbf{W}_n = [\boldsymbol{w}_1, \ldots, \boldsymbol{w}_n]$ ▷ periodic update for sparse representation weight matrix
15: $\mathbf{D}_n^{(i)} = \mathbf{X}_n \mathbf{W}_n{}^{\dagger}$ ▷ periodic dictionary update step
16: **end if** ▷ periodic update finishes
17: **end for** ▷ end of time iteration
18: **end for** ▷ end of iteration

stants. In Alg.4, steps 10-16 outline this periodic update of the instantaneous dictionary estimate. The check in step 10 makes sure that the periodic update is performed only every P time iterations. Steps 11-13 calculate the sparse representation for all the previously seen data vectors $t = 1, 2, \ldots, n$. Step 15 calculates the least squares solution for the dictionary estimate based on all the data vectors up to and including time n. λ in Alg.4 is the exponential weighting factor, utilized for forgetting the data vectors in the distant past [18]. By realizing the MOD-CU update step only with period P, we ensure that the computational complexity of the PURE algorithm does not get as high as the MOD-CU algorithm.

5. Computational Complexity Comparison

In this section we will discuss the computational complexities of the various DLA's in terms of the number of multiplications required. The common step in all the DLAs is the sparse representation step. If this sparse representation step is realized using OMP, its complexity is $\mathcal{O}((S^2 + M)K)$. S is the sparsity of the data vectors, M is data vector length and K is dictionary size. Assuming $S^2 \sim M$ and $M \sim K$, the sparse representation step complexity becomes $\mathcal{O}(K^2)$. Since sparse representation is the dominating step, for a complete iteration the asymptotic complexities of MOD, K-SVD and RLS-DLA are all equal to $\mathcal{O}(K^2 N)$, where N is the time duration of observed data [18]. MOD, K-SVD and RLS-DLA all solve a single sparse representation problem per time index, whereas MOD-CU requires the solution to n sparse representation problems at time index n. Hence, for a complete iteration the complexity for the MOD with coefficient update becomes $\mathcal{O}(K^2 \sum_{n=1}^{N} n) = \mathcal{O}(K^2 N^2)$. The asymptotic complexity for the MOD-CU is proportional to N^2, instead of N as is the case for MOD, K-SVD and RLS-DLA.

For a complete iteration PURE-DLA employs the solutions to $\big(N + P\sum_{m=1}^{N/P} m\big)$ sparse representation problems , where P is the period for the full update. Consequently, the multiplicational complexity for PURE-DLA becomes $\mathcal{O}(K^2N^2/P)$. If we ensure that $N/P = c$, where c is a constant, the complexity for PURE-DLA becomes $\mathcal{O}(cK^2N)$. Hence, the order of growth for the complexity of PURE-DLA is similar to MOD, K-SVD and RLS-DLA, as long as the ratio N/P stays constant.

6. Simulation Results

In this section we present experiments which detail the dictionary learning performance of the introduced algorithms when compared to DLAs from literature. We analyze the dictionary learning performance of the various algorithms under different signal-to-noise ratio (SNR) values. We also examine the performance of the PURE algorithm when utilized in image denoising. In the simulations of this section we have made use of the K-SVD implementations provided by the authors of [12][1].

6.1. Synthetic Experiments

In the first experiment set we analyze the dictionary learning performance of the algorithms in a synthetic setting. For each experiment a dictionary matrix of size $M \times K = 20 \times 30$ is generated randomly. The columns of the dictionary matrix which constitute the atoms of the dictionary are normalized to unit norm. The sparse representation matrix $\mathbf{W}$ is generated as to possess

[1]http://www.cs.technion.ac.il/ ronrubin/software.html

sparse columns with $S = 3$ nonzero elements at random locations, where the nonzero element values are taken from a zero mean, unit variance normal distribution. The number of generated data vectors is $N = 300$. White Gaussian observation noise is added to the data vectors as to result in desired SNR values. In addition to the algorithms MOD-CU and PURE introduced in this paper, we realize the online RLS-DLA [18], and the batch methods, MOD [15] and K-SVD [12]. The experiments are repeated 80 times for each setting, where a new dictionary is generated for each trial. A total of 60 iterations are realized over the data set for each trial. MOD-CU algorithm realizes Alg. 3 only in its initial iteration, where the remaining iterations realize the RLS-DLA. The forgetting factor is held to be constant with $\lambda = 0.99$ for PURE-DLA and RLS-DLA. For PURE-DLA the update period is $P = 60$. Hence, the dictionary update step is realized after every 60 time points. In all algorithms the sparse representation step is realized via OMP with a sparsity level of $S = 3$.

Fig.1 displays the number of correctly identified dictionary atoms out of $K = 30$ atoms for each of the algorithms at the end of the 60 iterations. The SNR changes from 10 dB up to 50 dB in steps of 10 dB. The criterion applied in determining whether an atom is correctly identified is the distance criterion defined in [12].

$$1 - |\boldsymbol{d}_i \hat{\boldsymbol{d}}_j| \tag{11}$$

The measure in (11) defines a distance between two normalized atoms, $\boldsymbol{d}_i$ from original generating dictionary and $\hat{\boldsymbol{d}}_j$ from learned dictionary. An atom is decided to be successfully identified when its distance from a learned atom is less than 0.01. The 80 trials are ordered based on the number of successfully

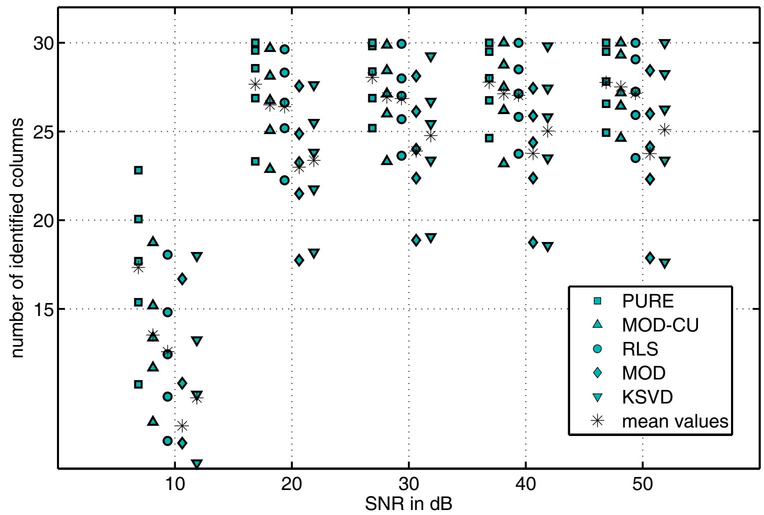

Figure 1: Number of correctly identified atoms after 60 iterations for PURE-DLA, MOD-CU, RLS-DLA, K-SVD and MOD using different SNR values. For each algorithm and for each SNR value there are a total of 80 trials grouped in five sets of trials.

identified atoms, and the mean number of identified atoms in five groups of experiments are presented in the scatter plot. The overall means of the 80 trials are also plotted. The results show that the PURE-DLA has the best atom identification performance among the algorithms for all SNR values. MOD-CU followed by RLS-DLA iterations is marginally better than RLS-DLA. Batch algorithms K-SVD and MOD have worse performance than the online algorithms.

In Fig.2 we present the evolution of the dictionary atom identification performance of the algorithms as a function of iteration number. The mean of the percent of correctly identified atoms is plotted versus the iteration index. The graphs show the results for 10 dB, 20 dB and 30 dB SNR. The batch algorithms K-SVD and MOD have poorer convergence than the online

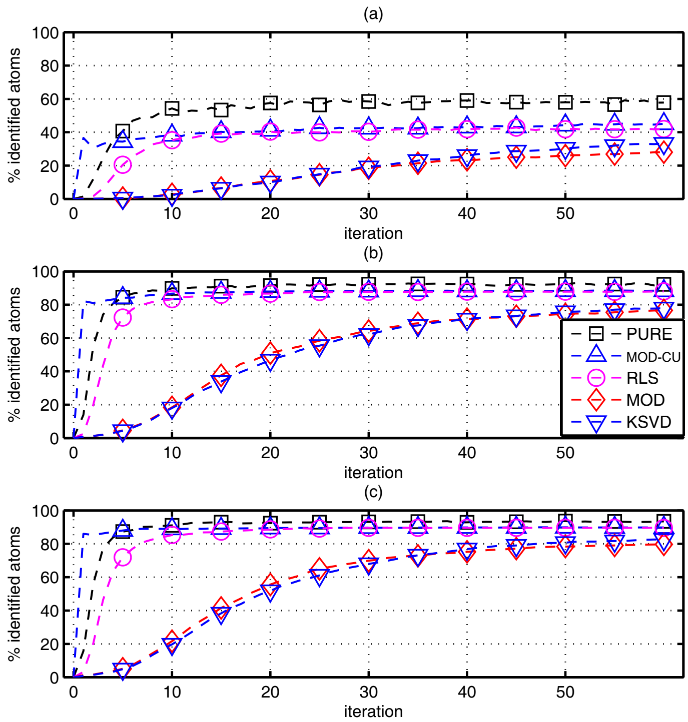

Figure 2: Percent of correctly identified atoms after each iteration averaged over 80 trials for PURE-DLA, MOD-CU, RLS-DLA, K-SVD and MOD. SNR values are (a) 10 dB, (b) 20 dB and (c) 30 dB.

algorithms. They do not fully converge after 60 iterations. The RLS-DLA algorithm has a gradual increase with iteration index and converges roughly after 20 iterations. For MOD-CU algorithm, the initial and computationally expensive MOD-CU iteration results in a steep increase in the number of identified atoms, jumping to almost the final correct identification performance in a single iteration. The curve for the MOD-CU converges to final values similar to those of the RLS-DLA. The convergence for MOD-CU supplemented by RLS-DLA takes less iterations than RLS-DLA alone. The PURE-DLA has the best convergence performance among the algorithms. It converges

Table 1: Computational time requirements of various DLAs for a single iteration.

Algorithm	PURE	MOD-CU	K-SVD	MOD	RLS
Time (sec)	0.6113	18.0019	0.2689	0.1201	0.1692

faster than RLS-DLA , and it also converges to a higher percent value than RLS-DLA. These results confirm that PURE-DLA presents a viable tradeoff between the high performance however computationally expensive MOD-CU algorithm and the RLS-DLA. The additional computational expense incurred by PURE-DLA over RLS-DLA is not prohibitive, since the update period of PURE-DLA is quite large with $N/P = 5$.

Table 1 details the computation time requirements of the algorithms compared in the above given synthetic dictionary identification experiment. In this table the computation time requirement for a single iteration of the D-LAs are listed. In Table 1, it is observed that the MOD-CU algorithm has much higher computation time requirement than the other algorithms, as it repeats the sparse representation update continuously. On the other hand, the PURE algorithm provides a good tradeoff by decreasing the computational requirement of the MOD-CU algorithm considerably using only periodic updates.

6.2. Image Denoising

There have been numerous attempts at image processing applications of sparse representation and dictionary learning methods [27, 28, 29, 30, 31]. In this section we apply the dictionary learning algorithms to the image denoising problem. In [27] dictionary learning using K-SVD is considered for the sparse representation of image patches. The sparse representation over the learned dictionary enables the development of a denoising algorithm which provides state-of-the art performance [28, 32]. We employ the novel PURE dictionary learning algorithm and also K-SVD and RLS-DLA algorithms in this image denoising framework introduced in [27]. Here, the dictionary learning algorithms are utilized for building an image patch dictionary from the patches of the noisy image. We also realize a version of the denoising framework which utilizes a fixed overcomplete DCT dictionary rather than learning a dictionary from the the noisy image itself. In the denoising experiments we utilize the general setup as introduced in [27]. The dictionary size is 64×256, where the atom length is chosen to handle 8×8 image patches. The dictionary is learned from a total of 5000 overlapping patches generated from the noisy image. The sparse representation step is realized by OMP. Observation noise with varying variance σ^2 is added, and σ is assumed to be a priori known. The terminating error threshold for OMP is chosen to be $\epsilon = 1.15\sigma$, and the averaging constant is $30/\sigma$ [27]. The forgetting factor is held to be constant with $\lambda = 0.99$ for PURE-DLA and RLS-DLA. The number of iterations for K-SVD is 10. RLS-DLA and PURE-DLA are run for 2 iterations. For PURE-DLA the update period is $P = 50$, that is the dictionary update step is realized after every 50 patches.

Table 2 summarizes the results in image denoising using dictionaries learned via PURE, K-SVD, RLS-DLA and additionally the fixed DCT dictionary. The reported results are averages over five experiments with different realizations of the noise. The experiments are realized for four different images which have also been used in the literature [27]. The results in Table 2

Table 2: Denoising results for different images and for differing PSNRs in decibels. In each cell four results are reported. Top left: K-SVD. Bottom left: RLS-DLA. Top right: PURE. Bottom right: Overcomplete DCT. In each cell the best result is boldfaced.

σ/PSNR	Lena		Barbara		Boats		House	
10/28.13	35.35	**35.41**	34.12	**34.18**	33.54	**33.58**	35.73	**35.90**
	33.89	35.29	32.40	33.95	32.58	33.43	33.90	35.39
15/24.61	33.48	**33.51**	31.99	**32.00**	31.55	**31.60**	34.04	**34.18**
	31.83	33.39	29.82	31.64	30.29	31.37	31.84	33.50
20/22.11	32.11	**32.14**	**30.41**	30.36	30.06	**30.11**	32.76	**32.89**
	30.60	31.99	28.14	29.94	28.79	29.91	30.56	32.10
25/20.17	**31.03**	31.02	**29.13**	29.09	28.96	**29.02**	**31.75**	31.56
	29.65	30.90	26.91	28.62	27.73	28.77	29.46	30.98
50/14.15	**27.58**	27.52	**24.86**	24.78	**25.64**	25.62	**27.69**	27.61
	26.79	27.44	23.78	24.74	24.70	25.60	26.65	27.51

suggest that using the coefficient update enhanced PURE algorithm in place of RLS-DLA improves the denoising performance. The dictionaries learned using K-SVD and PURE in general have similar performance in this denoising setting.They perform in general better than the dictionaries learned by RLS-DLA and the fixed DCT dictionary. Hence, utilizing a dictionary learned via PURE with the coefficient update results in a performance gain when compared to the RLS-DLA. Figs.3 and 5 present the original, noisy and denoised "Barbara" and "Boat" images for noise level $\sigma = 20$. Figs.4 and 6 show two scaled up sections of the images presented in Figs.3 and 5 to display the denoising performance in more detail. The detail figures in Figs. 4 and 6 demonstrate that the PURE-DLA and K-SVD dictionaries are better than the RLS-DLA dictionary in keeping the details of the original image.

7. Conclusions

We have presented a new dictionary learning algorithm for the online dictionary training from data. Firstly, we considered a coefficient update improvement on the MOD algorithm, which we called as MOD-CU algorithm. This method provides a computationally expensive but improved variation on the MOD algorithm. Secondly we propose an algorithm which provides a compromise between the computationally expensive full MOD-CU iteration and the RLS-DLA, and we call this method as the PURE algorithm. The new PURE algorithm introduces a novel periodic sparse representation update procedure into the two-step iterative dictionary learning approach. Simulation results for synthetic dictionary identification scenario depict the high convergence performance of MOD-CU. The results also illustrate the superior convergence and estimation performance of the PURE over RLS-DLA and MOD. We also implemented the PURE algorithm in the image denoising setting. The image denoising results confirm that PURE algorithm provides a computationally tractable enhancement for the RLS-DLA. Utilizing a dictionary learned from the noisy image via PURE, gives in general better results than using RLS-DLA or employing a fixed overcomplete

DCT dictionary. PURE algorithm presents an improved online approach for high performance dictionary learning without compromising computational complexity. One possible future research topic might be the introduction of the new sparse representation update procedure into structured or group-sparse dictionary learning algorithms. A second research direction would be using this strategy in the recently introduced analysis sparsity and analysis operator learning algorithms.

References

[1] R. Rubinstein, A. M. Bruckstein, and M. Elad, "Dictionaries for sparse representation modeling," *Proc. IEEE*, vol. 98, no. 6, pp. 1045–1057, 2010.

[2] M.D. Plumbley, T. Blumensath, L. Daudet, R. Gribonval, and M.E. Davies, "Sparse representations in audio and music: From coding to source separation," *Proc. IEEE*, vol. 98, no. 6, pp. 995 –1005, June 2010.

[3] Wooyoung Kim, Bernard Chen, Jingu Kim, Yi Pan, and Haesun Park, "Sparse nonnegative matrix factorization for protein sequence motif discovery," *Expert Systems with Applications*, vol. 38, no. 10, pp. 13198 – 13207, 2011.

[4] J.A. Tropp and S.J. Wright, "Computational methods for sparse solution of linear inverse problems," *Proc. IEEE*, vol. 98, no. 6, pp. 948 –958, 2010.

[5] S. S. Chen, D. L. Donoho, and M. A. Saunders, "Atomic decomposition by basis pursuit," *SIAM J. Sci. Comput.*, vol. 20, no. 1, pp. 33–61, 1998.

[6] J. A. Tropp and A. C. Gilbert, "Signal recovery from random measurements via orthogonal matching pursuit," *IEEE Trans. Inf. Theory*, vol. 53, no. 12, pp. 4655–4666, Dec. 2007.

[7] I. Tǒsić and P. Frossard, "Dictionary learning," *IEEE Signal Process. Mag.*, vol. 28, no. 2, pp. 27 –38, Mar. 2011.

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

[9] 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, July 2010.

[10] B.A. Olshausen and D.J. Field, "Sparse coding with an overcomplete basis set: A strategy employed by v1?," *Vision Research*, vol. 37, no. 23, pp. 3311–3325, 1997.

[11] Kenneth Kreutz-Delgado, Joseph F. Murray, Bhaskar D. Rao, Kjersti Engan, Te-Won Lee, and Terrence J. Sejnowski, "Dictionary learning algorithms for sparse representation," *Neural Comput.*, vol. 15, no. 2, pp. 349–396, 2003.

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

[13] A. Rakotomamonjy, "Direct optimization of the dictionary learning problem," *Signal Processing, IEEE Transactions on*, vol. 61, no. 22, pp. 5495–5506, 2013.

[14] K. Engan, S.O. Aase, and J.H. Husøy, "Method of optimal directions for frame design," in *Proc. ICASSP*, 1999, vol. 5, pp. 2443–2446.

[15] K. Engan, K. Skretting, and J. H. Husøy, "Family of iterative LS-based dictionary learning algorithms, ILS-DLA, for sparse signal representation," *Digit. Signal Process.*, vol. 17, no. 1, pp. 32–49, 2007.

[16] M. Sadeghi, M. Babaie-Zadeh, and C. Jutten, "Dictionary learning for sparse representation: A novel approach," *Signal Processing Letters, IEEE*, vol. 20, no. 12, pp. 1195–1198, 2013.

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

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

[19] B. Babadi, N. Kalouptsidis, and V. Tarokh, "SPARLS: The sparse RLS algorithm," *IEEE Trans. Signal Process.*, vol. 58, no. 8, pp. 4013–4025, Aug. 2010.

[20] E. M. Eksioglu and A. K. Tanc, "RLS algorithm with convex regularization," *IEEE Signal Process. Lett.*, vol. 18, no. 8, pp. 470 –473, aug. 2011.

[21] J. Mairal, F. Bach, J. Ponce, and G. Sapiro, "Online learning for matrix factorization and sparse coding," *Journal of Machine Learning Research*, vol. 11, pp. 19–60, 2010.

[22] Shanshan Wang, Qiegen Liu, Yong Xia, Pei Dong, Jianhua Luo, Qiu Huang, and David Dagan Feng, "Dictionary learning based impulse noise removal via L1-L1 minimization," *Signal Processing*, vol. 93, no. 9, pp. 2696 – 2708, 2013.

[23] Shuyuan Yang, Linfang Zhao, Min Wang, Yueyuan Zhang, and Licheng Jiao, "Dictionary learning and similarity regularization based image noise reduction," *Journal of Visual Communication and Image Representation*, vol. 24, no. 2, pp. 181 – 186, 2013.

[24] Qiegen Liu, Shanshan Wang, Jianhua Luo, Yuemin Zhu, and Meng Ye, "An augmented Lagrangian approach to general dictionary learning for image denoising," *Journal of Visual Communication and Image Representation*, vol. 23, no. 5, pp. 753–766, 2012.

[25] Zhuolin Jiang, Zhe Lin, and L.S. Davis, "Label consistent k-svd: Learning a discriminative dictionary for recognition," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 35, no. 11, pp. 2651–2664, 2013.

[26] K. Kreutz-Delgado, J. F. Murray, B. D. Rao, K. Engan, T. W. Lee, and T. J. Sejnowski, "Dictionary learning algorithms for sparse representation," *Neural Comput.*, vol. 15, no. 2, pp. 349–396, 2003.

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

[28] M. Aharon and M. Elad, "Sparse and redundant modeling of image content using an image-signature-dictionary," *SIAM J. Img. Sci.*, vol. 1, no. 3, pp. 228–247, 2008.

[29] M. Elad, M. A. T. Figueiredo, and Yi Ma, "On the role of sparse and redundant representations in image processing," *Proc. IEEE*, vol. 98, no. 6, pp. 972–982, 2010.

[30] J. M. Duarte-Carvajalino and G. Sapiro, "Learning to sense sparse signals: Simultaneous sensing matrix and sparsifying dictionary optimization," *IEEE Trans. Image Process.*, vol. 18, no. 7, pp. 1395–1408, 2009.

[31] P. Chatterjee and P. Milanfar, "Clustering-based denoising with locally learned dictionaries," *IEEE Trans. Image Process.*, vol. 18, no. 7, pp. 1438 –1451, july 2009.

[32] G. Peyré, "A review of adaptive image representations," *IEEE J. Sel. Topics Signal Process.*, vol. 5, no. 5, pp. 896–911, sept 2011.

Figure 3: Image denoising experiment results for image "Barbara" with $\sigma = 20$. a) Noisy image with $\sigma = 20$ (22.11 dB). b) Image denoised using dictionary learned via RLS-DLA (28.14 dB). c) Image denoised using dictionary learned via K-SVD (30.41 dB). d) Image denoised using dictionary learned via PURE (30.36 dB).

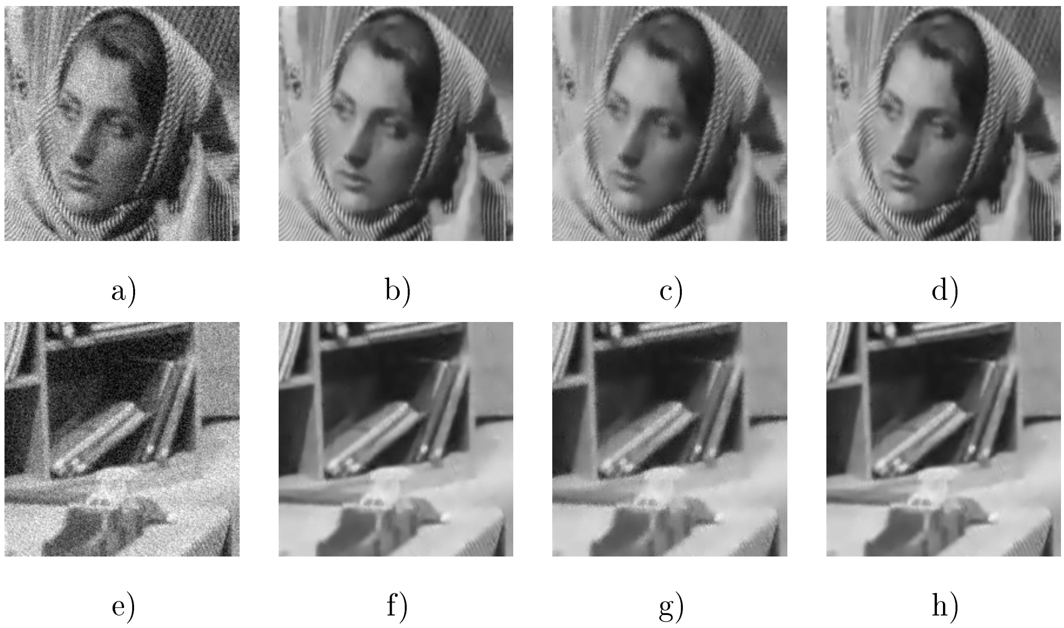

Figure 4: Scaled up comparison of denoising results from Fig.3. a), e) Noisy images. b), f), Images denoised using dictionary learned via PURE. c), g) Images denoised using dictionary learned via RLS-DLA. d), h) Images denoised using dictionary learned via K-SVD.

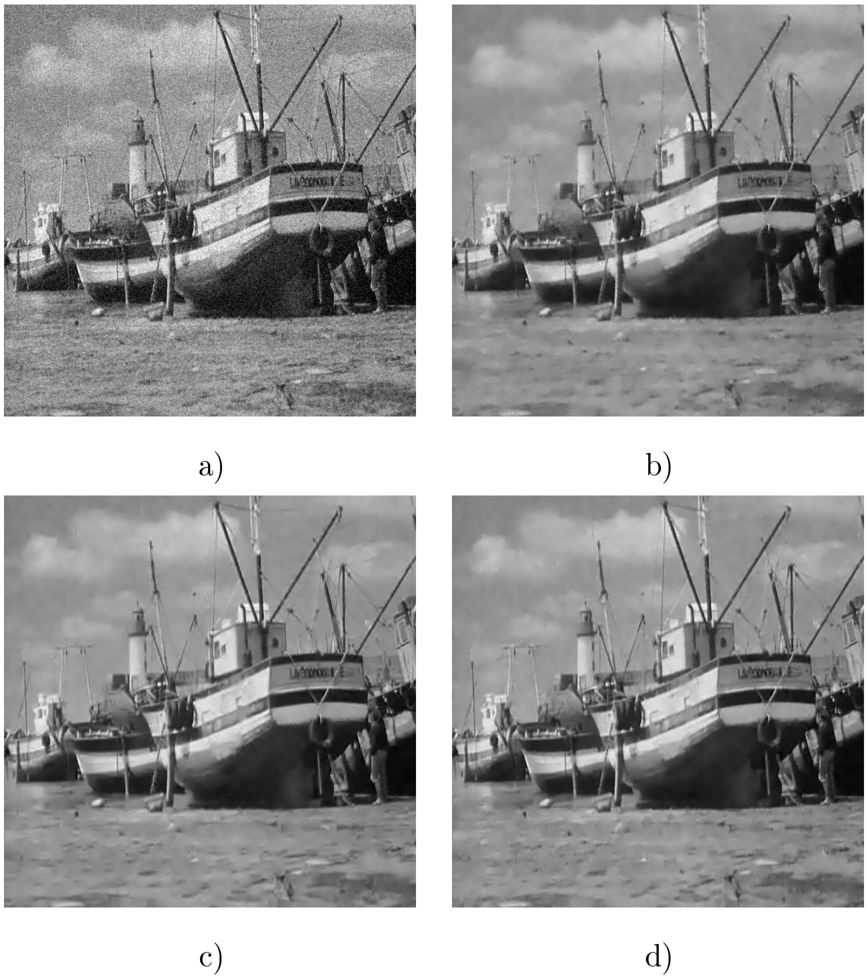

Figure 5: Image denoising experiment results for image "Boat" with $\sigma = 20$. a) Noisy image with $\sigma = 20$ (22.11 dB). b) Image denoised using dictionary learned via RLS-DLA (28.79 dB). c) Image denoised using dictionary learned via K-SVD (30.06 dB). d) Image denoised using dictionary learned via PURE (30.11 dB).

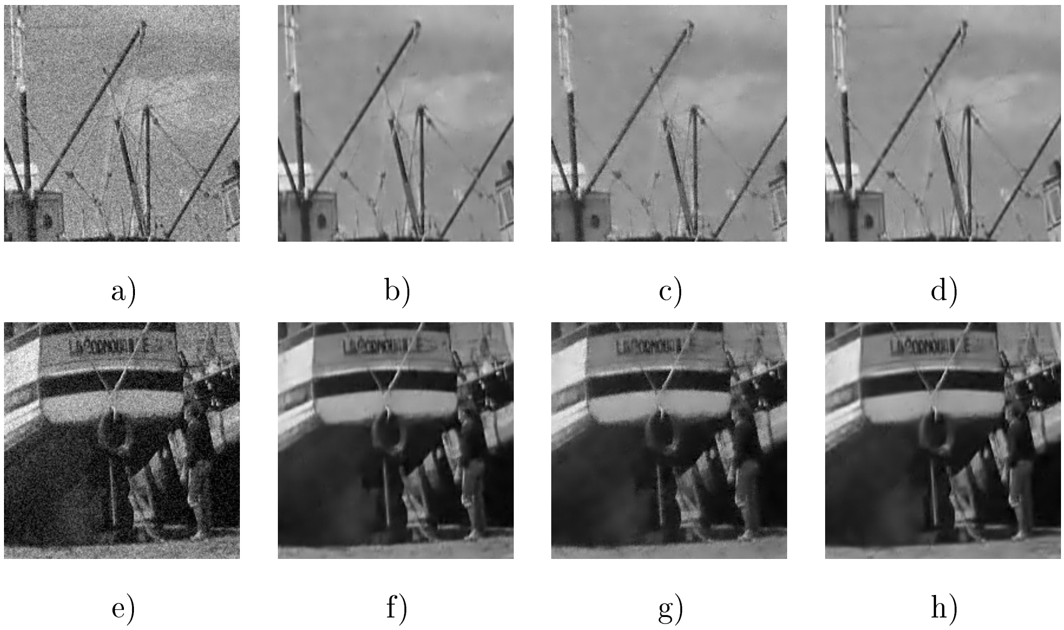

Figure 6: Scaled up comparison of denoising results from Fig.5. a), e) Noisy images. b), f), Images denoised using dictionary learned via PURE. c), g) Images denoised using dictionary learned via RLS-DLA. d), h) Images denoised using dictionary learned via K-SVD.