# New Dictionary Learning Methods for Two-Dimensional Signals

Firooz Shahriari-Mehr, Javad Parsa, Massoud Babaie-Zadeh
*Electrical Engineering Department*
*Sharif University of Technology, Tehran, Iran*
firoozshahriari@gmail.com, javadparsa86@gmail.com, mbzadeh@sharif.edu

Christian Jutten
*GIPSA-Lab*
*Univ. Grenoble Alpes, Grenoble, France*
Christian.Jutten@gipsa-lab.grenoble-inp.fr

***Abstract*—By growing the size of signals in one-dimensional dictionary learning for sparse representation, memory consumption and complex computations restrict the learning procedure. In applications of sparse representation and dictionary learning in two-dimensional signals (e.g. in image processing), if one opts to convert two-dimensional signals to one-dimensional ones, and use the existing one-dimensional dictionary learning and sparse representation techniques, too huge signals and dictionaries will be encountered. Two-dimensional dictionary learning has been proposed to avoid this problem. In this paper, we propose two algorithms for two-dimensional dictionary learning. According to our simulations, the proposed algorithms have noticeable improvement in both convergence rate and computational load in comparison to one-dimensional methods.**

***Index Terms*—Two-dimensional dictionary learning, sparse representation, 2D Signals, convex approximation.**

## I. Introduction

Dictionary learning for sparse representation [1], [2] has many applications in signal and image processing. In one-dimensional (1D) sparse representation, a signal $\mathbf{y} \in \mathbb{R}^n$ is to be represented as a linear combination of some basic signals $\mathbf{d}_1, \ldots, \mathbf{d}_m$, where $m > n$ and $\forall i, \mathbf{d}_i \in \mathbb{R}^n$. As suggested in [3], $\mathbf{d}_i$'s are called atoms, and their collection is called *dictionary*, which can be represented by the matrix $\mathbf{D} \in \mathbb{R}^{n \times m}$, composed of all atoms as its columns. Finding the sparse representation of a known signal $\mathbf{y}$ over the dictionary $\mathbf{D}$ requires finding the sparse solution of the under-determined system of linear equations $\mathbf{y} = \mathbf{D}\mathbf{x}$, that is, solving

$$\min_{\mathbf{x}} \|\mathbf{x}\|_0 \quad s.t. \quad \mathbf{y} = \mathbf{D}\mathbf{x}, \tag{1}$$

where $\|\mathbf{x}\|_0$ denotes the $\ell^0$ (pseudo-) norm, that is, the number of non-zeros entries of the vector $\mathbf{x}$. This problem, called sparse coding, is NP-hard, and many algorithms have been proposed to estimate its solution, *e.g.* [3]–[7] to name a few.

In the above problem, the dictionary may be chosen from fixed dictionaries such as wavelet and Over-complete Discrete Cosine Transform (ODCT) dictionaries. However, to have a better (i.e. sparser) representation for a class of signals, one may opt to *learn* a dictionary for that class of signals. In *Dictionary Learning* (DL) problem, a set of training signals $\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_L$ is available and a dictionary $\mathbf{D}$ is to be learned such that it results in the sparsest representation for this set of training signals. So, defining $\mathbf{Y} \triangleq [\mathbf{y}_1, \mathbf{y}_2, \ldots \ \mathbf{y}_L] \in \mathbb{R}^{n \times L}$ the dictionary learning problem is expressed as

$$(\mathbf{D}^*, \mathbf{X}^*) = \underset{\mathbf{D} \in \mathcal{D}, \mathbf{X} \in \mathcal{X}}{\operatorname{argmin}} \ \|\mathbf{Y} - \mathbf{D}\mathbf{X}\|_F^2, \tag{2}$$

where $\mathcal{D} \triangleq \left\{\mathbf{D}: \ \forall i, \|\mathbf{d}_i\|_2^2 = 1\right\}$, $\mathcal{X} \triangleq \{\mathbf{X}: \ \forall i, \|\mathbf{x}_i\|_0 \leq \tau\}$ and $\mathbf{X} \in \mathbb{R}^{m \times L}$. Many dictionary learning algorithms have been proposed in the literature, *e.g.* [8]–[14]. Most of them use alternating minimization over $\mathbf{D}$ and $\mathbf{X}$ to solve (2), so, each iteration of them is composed of a sparse representation stage (in which $\mathbf{D}$ is kept fixed) and a dictionary update stage (in which $\mathbf{X}$ is kept fixed). Some of them differ only in the second stage, for instance, Method of Optimal Directions (MOD) [9] use gradient-projection approach to find the dictionary, and K-SVD [8] updates the atoms consecutively by using the SVD decomposition. In [10], a new jointly convex objective function has been obtained for dictionary learning by using first order series expansion instead of the term $\mathbf{D}\mathbf{X}$ in (2).

In decomposition of two-dimensional (2D) signals, a signal $\mathbf{Y} \in \mathbb{R}^{n_1 \times n_2}$ is to be represented as a linear combination of 2D atoms $\mathbf{\Phi}_{ij}$, i.e. $\mathbf{Y} = \sum_{i=1}^{m_1} \sum_{j=1}^{m_2} x_{ij} \mathbf{\Phi}_{ij}$. However, most 2D atoms that are used in image processing (*e.g.* Fourier atoms) are separable [15, Chapter 2], meaning that there are vectors $\mathbf{a}_i \in \mathbb{R}^{n_1}$ and $\mathbf{b}_j \in \mathbb{R}^{n_2}$ such that $\mathbf{\Phi}_{ij} = \mathbf{a}_i \mathbf{b}_j^T$, $1 \leq i \leq m_1$, $1 \leq j \leq m_2$. Separable atoms are used in sparse coding [16] and dictionary learning [17] to represent images or learn a dictionary for a set of training signals. This separable structure lets 2D signals to be represented as $\mathbf{Y} = \mathbf{A}\mathbf{X}\mathbf{B}^T$ [16], where $\mathbf{A} = [\mathbf{a}_1, \mathbf{a}_2, ..., \mathbf{a}_{m_1}] \in \mathbb{R}^{n_1 \times m_1}$, $\mathbf{B} = [\mathbf{b}_1, \mathbf{b}_2, ..., \mathbf{b}_{m_2}] \in \mathbb{R}^{n_2 \times m_2}$ and $\mathbf{X} \in \mathbb{R}^{m_1 \times m_2}$ is the representation of the signal. The 1D equivalent of this expression is vec($\mathbf{Y}$) = $\mathbf{D}$vec($\mathbf{X}$), where 'vec' of a matrix stands for the vector obtained by stacking its columns, and $\mathbf{D}$ is a Kronecker product of $\mathbf{A}$ and $\mathbf{B}$, i.e. $\mathbf{D} = \mathbf{B} \otimes \mathbf{A}$ [16], [18]. A trivial method to find the sparse representation of 2D signals is to convert them to 1D signals, then use the proposed methods for 1D sparse coding. This approach results in huge dictionaries specially for large 2D signals that requires a tremendous amount of memory and computational load. For example, finding the sparse representation of a 2D signal $\mathbf{Y} \in \mathbb{R}^{40 \times 50}$ on dictionaries $\mathbf{A} \in \mathbb{R}^{40 \times 100}$ and $\mathbf{B} \in \mathbb{R}^{50 \times 100}$ results in the Kronecker dictionary $\mathbf{D} \in \mathbb{R}^{2000 \times 10000}$, which

This work has been partially supported by Research Office of Sharif University of Technology, and by the ERC project CHESS: 2012-ERC-AdG-320684.

is practically inefficient. The Kronecker dictionary requires a memory of order $\mathcal{O}(n_1n_2m_1m_2)$, while dictionaries with separable structure need a memory of order $\mathcal{O}(n_1m_1+n_2m_2)$.

To avoid the above problems, the authors of [16] have proposed to use 2D sparse representation as

$$\min_{\mathbf{X}} \|\mathbf{X}\|_0 \quad s.t. \quad \mathbf{Y}=\mathbf{A}\mathbf{X}\mathbf{B}^T. \tag{3}$$

To solve the above problem, the algorithms 2D-SL0 [16] and 2D-OMP [19] have already been proposed in the literature.

In dictionary learning, when dealing with 2D signals, the same problems exists, so the 2D dictionary learning problem for a set of 2D signals $\mathcal{Y}=(\mathbf{Y}_1,\mathbf{Y}_2,...,\mathbf{Y}_L)$ can be written as

$$(\mathbf{A}^*,\mathbf{X}^*,\mathbf{B}^*)=\underset{\mathbf{X}_i\in\mathcal{X}_i,\mathbf{A}\in\mathcal{A},\mathbf{B}\in\mathcal{B}}{\operatorname{argmin}} \sum_{i=1}^{L}\|\mathbf{Y}_i-\mathbf{A}\mathbf{X}_i\mathbf{B}^T\|_F^2, \tag{4}$$

where $\mathcal{X}_i \triangleq \{\mathbf{X}_i : \|\mathbf{X}_i\|_0 \le \tau\}$, $\mathcal{A} \triangleq \big\{\mathbf{A}:\ \forall i, \|\mathbf{a}_i\|_2^2=1\big\}$ and $\mathcal{B} \triangleq \big\{\mathbf{B}:\ \forall i, \|\mathbf{b}_i\|_2^2=1\big\}$. The first constraint imposes the sparsity of signal representations and the other ones avoid scaling ambiguity of dictionaries.

To solve (4), the authors of [17] have defined two regularization terms for measuring sparsity of signals and mutual coherence of the dictionaries, and by utilizing optimization on matrix manifolds, have proposed an algorithm called Separable Dictionary Learning (SeDiL).

In this paper, two new algorithms will be proposed to solve 2D dictionary learning; besides, a new jointly convex objective function will be achieved for it. The proposed methods result in suitable recovery percentage for dictionary, with faster convergence rate and much less computational load and memory consumption. Moreover, 2D algorithms can recover the dictionary with few training signals, unlike 1D algorithms. SeDiL cannot recover the known dictionary and fails in the experiments. Moreover, we study the proposed algorithms in image denoising application.

The rest of this paper is organized as follows. Section II presents the main ideas and our proposed methods. Then, Section III evaluates the new algorithms numerically.

## II. The Proposed Methods

In this section, two new algorithms are proposed to solve the 2D dictionary learning problem.

### A. 2D-MOD

Problem (4) is jointly non-convex over $\mathbf{A}$, $\mathbf{B}$ and $\mathbf{X}_i$, so we can use alternating minimization to solve it. This approach reduces to alternations between sparse coding for 2D training signals where dictionaries $\mathbf{A}$ and $\mathbf{B}$ are kept fixed, and then updating dictionaries $\mathbf{A}$ and $\mathbf{B}$. This method is based on MOD [9] algorithm for 1D dictionary learning, so we call the proposed algorithm 2D-MOD, which has three steps:

1) Sparse representations: By keeping the dictionaries $\mathbf{A}$ and $\mathbf{B}$ fixed, the objective function (4) becomes

$$\mathbf{X}_i^{(k+1)}=\underset{\mathbf{X}_i\in\mathcal{X}_i}{\operatorname{argmin}} \sum_{i=1}^{L}\left\|\mathbf{Y}_i-\mathbf{A}\mathbf{X}_i\mathbf{B}^T\right\|_F^2, \tag{5}$$

which is a usual 2D sparse representation for each training signal, and could be solved by 2D-SL0 [16] or 2D-OMP [19].

2) $\mathbf{A}$-update: The problem of updating dictionary $\mathbf{A}$ is:

$$\mathbf{A}^{(k+1)}=\underset{\mathbf{A}\in\mathcal{A}}{\operatorname{argmin}} \sum_{i=1}^{L}\left\|\mathbf{Y}_i-\mathbf{A}\mathbf{X}_i^{(k+1)}\mathbf{B}^T\right\|_F^2. \tag{6}$$

Gradient Projection (GP) is used to solve (6). The final equation for updating $\mathbf{A}$ is as follows

$$\text{normalize}\Big\{\Big(\sum_{i=1}^{L}\mathbf{Y}_i\mathbf{B}\mathbf{X}_i^T\Big)\Big(\sum_{i=1}^{L}\mathbf{X}_i\mathbf{B}^T\mathbf{B}\mathbf{X}_i^T\Big)^{-1}\Big\}, \tag{7}$$

where 'normalize' stands for a matrix that all its columns have unit Euclidean norm.

3) $\mathbf{B}$-update: The problem is very similar to (6), and the final equation after applying GP is

$$\text{normalize}\Big\{\Big(\sum_{i=1}^{L}\mathbf{Y}_i^T\mathbf{A}\mathbf{X}_i\Big)\Big(\sum_{i=1}^{L}\mathbf{X}_i^T\mathbf{A}^T\mathbf{A}\mathbf{X}_i\Big)^{-1}\Big\}. \tag{8}$$

### B. 2D-CMOD

In this subsection, a new jointly convex objective function is achieved for 2D dictionary learning by using the convexification idea, which is proposed in [10]. The proposed optimization approach is a gradient method based on a first order approximation. Writing $\mathbf{A}=\mathbf{A}_a+(\mathbf{A}-\mathbf{A}_a)$, $\mathbf{B}=\mathbf{B}_a+(\mathbf{B}-\mathbf{B}_a)$ and $\mathbf{X}=\mathbf{X}_a+(\mathbf{X}-\mathbf{X}_a)$, we have

$$\begin{aligned}&\mathbf{A}\mathbf{X}\mathbf{B}^T=\mathbf{A}_a\mathbf{X}_a\mathbf{B}^T+\mathbf{A}\mathbf{X}_a\mathbf{B}_a^T+\mathbf{A}_a\mathbf{X}\mathbf{B}_a^T-2\mathbf{A}_a\mathbf{X}_a\mathbf{B}_a^T+\\&\mathbf{A}_a(\mathbf{X}-\mathbf{X}_a)(\mathbf{B}-\mathbf{B}_a)^T+(\mathbf{A}-\mathbf{A}_a)\mathbf{X}_a(\mathbf{B}-\mathbf{B}_a)^T+\\&(\mathbf{A}-\mathbf{A}_a)(\mathbf{X}-\mathbf{X}_a)\mathbf{B}_a^T+(\mathbf{A}-\mathbf{A}_a)(\mathbf{X}-\mathbf{X}_a)(\mathbf{B}-\mathbf{B}_a)^T.\end{aligned} \tag{9}$$

The last four terms contain higher order differences, and become negligible if the first order differences, *i.e.* $(\mathbf{A}-\mathbf{A}_a)$, $(\mathbf{B}-\mathbf{B}_a)$ and $(\mathbf{X}-\mathbf{X}_a)$, are small. So

$$\mathbf{A}\mathbf{X}\mathbf{B}^T\approx\mathbf{A}_a\mathbf{X}_a\mathbf{B}^T+\mathbf{A}\mathbf{X}_a\mathbf{B}_a^T+\mathbf{A}_a\mathbf{X}\mathbf{B}_a^T-2\mathbf{A}_a\mathbf{X}_a\mathbf{B}_a^T. \tag{10}$$

By substituting (10) in (4), the following new convex dictionary learning problem is achieved

$$\begin{aligned}(\mathbf{A}^*,\mathbf{X}^*,\mathbf{B}^*)=\underset{\mathbf{X}_i\in\mathcal{X}_i,\mathbf{A}\in\mathcal{A},\mathbf{B}\in\mathcal{B}}{\operatorname{argmin}} \sum_{i=1}^{L}\|\mathbf{Y}_i+2\mathbf{A}_a\mathbf{X}_{a,i}\mathbf{B}_a^T\\-\mathbf{A}_a\mathbf{X}_{a,i}\mathbf{B}^T-\mathbf{A}\mathbf{X}_{a,i}\mathbf{B}_a^T-\mathbf{A}_a\mathbf{X}_i\mathbf{B}_a^T\|_F^2.\end{aligned} \tag{11}$$

To solve the above problem while maintaining the constraints, alternating minimization is used. The minimization is done over $\mathbf{A}$, $\mathbf{B}$ and $\mathbf{X}_i$, where $\mathbf{A}_a$, $\mathbf{B}_a$ and $\mathbf{X}_{a,i}$ represent previous values of these parameters. More precisely, the following three steps are repeated iteratively[1]:

[1]Actually, there are several scenarios to set the fixed parameters, as is discussed in [20] for the 1D problem. Our choice here is inspired from the choice #4 proposed in [20] for the 1D problem.

**Algorithm 1:** 2D-CMOD

**Input:** Signal set: $\mathcal{Y}$, Sparsity level: $s$, Number of training signals: $num\_train$, Algorithm iterations: $iter$.
**Output:** Sparse representations: $\mathbf{X}_i$'s, Dictionaries: $\mathbf{A}$ and $\mathbf{B}$.

Initialize dictionaries $\mathbf{A}$ and $\mathbf{B}$.
Set: $\mathbf{A}^{(0)} = \mathbf{A}^{(-1)} = \mathbf{A}, \mathbf{B}^{(0)} = \mathbf{B}^{(-1)} = \mathbf{B}$.
**for** $k = 0$ **to** $iter - 1$ **do**
  **for** $i = 1$ **to** $num\_train$ **do**
    $\mathbf{Z}_i = \mathbf{Y}_i - (\mathbf{A}^{(k)} - \mathbf{A}^{(k-1)})\mathbf{X}_i(\mathbf{B}^{(k-1)})^T - \mathbf{A}^{(k-1)}\mathbf{X}_i(\mathbf{B}^{(k)} - \mathbf{B}^{(k-1)})^T$
    $\mathbf{X}_i =$ Sparse Coding$(\mathbf{Z}_i, \mathbf{A}^{(k)}, \mathbf{B}^{(k)}, s)$
  **end for**
  $\mathbf{A}^{(k+1)}$ = Update dictionary $\mathbf{A}$ as in (7).
  $\mathbf{B}^{(k+1)}$ = Update dictionary $\mathbf{B}$ as in (8).
**end for**

1) Sparse representation: Let

$$\begin{cases} \mathbf{A}_a = \mathbf{A}^{(k-1)}, \mathbf{A} = \mathbf{A}^{(k)} \\ \mathbf{B}_a = \mathbf{B}^{(k-1)}, \mathbf{B} = \mathbf{B}^{(k)} \\ \mathbf{X}_a = \mathbf{X}^{(k)} \\ \mathbf{Z}_i = \mathbf{Y}_i - (\mathbf{A}^{(k)} - \mathbf{A}^{(k-1)})\mathbf{X}_i^{(k)}(\mathbf{B}^{(k-1)})^T \\ \qquad - \mathbf{A}^{(k-1)}\mathbf{X}_i^{(k)}(\mathbf{B}^{(k)} - \mathbf{B}^{(k-1)})^T, \end{cases}$$

where $k$ denotes the iteration number, and $\mathbf{Z}_i$ merges all the first four terms inside the norm in (11), which does not depend on the optimizing variables of this step ($\mathbf{X}_i$'s). Then, the sparse representation problem of step $k+1$ is

$$\mathbf{X}_i^{(k+1)} = \underset{\mathbf{X}_i \in \mathcal{X}}{\operatorname{argmin}} \sum_{i=1}^{L} \|\mathbf{Z}_i - \mathbf{A}^{(k-1)}\mathbf{X}_i(\mathbf{B}^{(k-1)})^T\|_F^2,$$

which is a usual 2D sparse coding for all $\mathbf{Z}_i$'s.

2) $\mathbf{A}$-update: Let

$$\begin{cases} \mathbf{X}_a = \mathbf{X} = \mathbf{X}^{(k+1)} \\ \mathbf{B}_a = \mathbf{B} = \mathbf{B}^{(k)} \end{cases}.$$

Then, by the above assumptions, dictionary $\mathbf{A}$ is updated using (7).

3) $\mathbf{B}$-update: Let

$$\begin{cases} \mathbf{X}_a = \mathbf{X} = \mathbf{X}^{(k+1)} \\ \mathbf{A}_a = \mathbf{A} = \mathbf{A}^{(k+1)} \end{cases}.$$

Then, by the above assumptions, dictionary $\mathbf{B}$ is updated using (8). We call the resulting algorithm 2D-CMOD (Convex MOD), and its pseudo-code is summarized in Algorithm 1.

## III. Simulation Results

In this section, the proposed methods are simulated on both synthetic and real data. For all algorithms, Orthogonal Matching Pursuit (OMP) [5] has been used as the sparse coding algorithm. As a rough measure of the complexities of the algorithms, their run times will be reported. Our simulations were performed in MATLAB 2018b environment on a system with 4.0 GHz CPU, and 16 GB RAM, under Microsoft Windows 10 64-bit operating system.

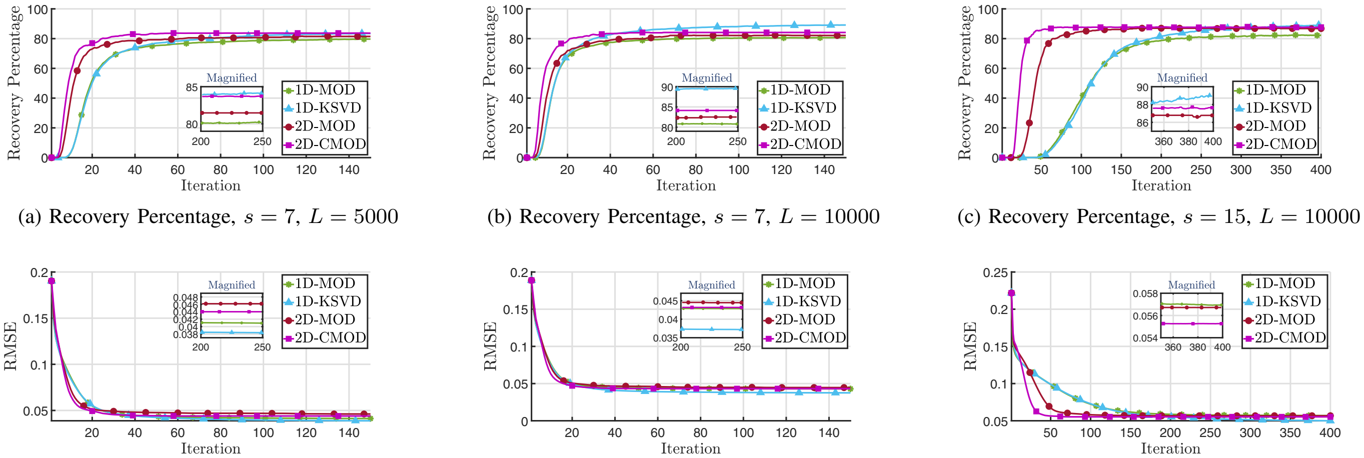
### *A. Successful Recovery of Known Dictionaries & RMSE*

In this test, 2D signals are assumed to be of size $n \times n$, where $n$ will take different values from 10 to 25 in our experiments. We generate two random dictionaries $\mathbf{A}$ and $\mathbf{B} \in \mathbb{R}^{n \times 2n}$ with zero mean and unit variance independent and identically distributed (i.i.d) Gaussian entries, followed by normalization. Sparse representation matrices, i.e. $\mathbf{X}_i$'s, are randomly produced with different sparsity levels $s$, i.e. $s$ non-zero elements. Then, a collection of $L$ training signals are generated as $\mathbf{Y}_i = \mathbf{A}\mathbf{X}_i\mathbf{B}^T + \mathbf{N}_i$, where $\mathbf{N}_i$ shows the additive white Gaussian noise with Signal to Noise Ratio (SNR) level of 30dB. Dictionary learning methods are applied to these signals by assuming that the sparsity level is known. For 1D methods, the vec($\mathbf{Y}_i$)'s are used as training signals. Successful recovery of the Kronecker dictionary $\mathbf{D}$, which could be computed as the ratio of successfully recovered atoms to the number of all atoms (an atom would be called successfully recovered if the correlation between the atom and the true one be more than 0.99 ), and RMSE, which is defined as RMSE $= \sqrt{\sum_{i=1}^{L} \|\mathbf{Y}_i - \mathbf{A}\mathbf{X}_i\mathbf{B}^T\|_F^2 \Big/ n^2 L}$, are used as the criteria of the efficiency of the algorithms.

Figure 1 shows the successful recovery percentage and RMSE for simulations with $n = 10$, $L = 500n, 1000n$ and $s = 7, 15$. All the reported values are averaged over 25 trials. SeDiL [17] is not included in the figure because it failed to recover the dictionary[2]. By comparing the Figs. 1a and 1b, we can say that 1D methods need more training signals than 2D methods. If the number of training signals is small, 1D methods' performance and their convergence rate will decrease. Moreover, we observe that when the number of training signals decreased from $500n$ to around $150n$, 1D methods failed to recover the dictionary, while our 2D algorithms could still recover it. The second observation, which can be seen from Figs. 1b and 1c is that when the sparsity level increases, the performance of 2D methods will improve, and the convergence rate of 1D methods will decrease. As the experiment's result, the proposed methods need much fewer training signals and perform better in simulations with high sparsity level.

To roughly measure the complexity of the algorithms, the average required time to recover 80 percent of the Kronecker dictionary is reported. In this experiment, the parameters are set as $s = n$ and $L = 1000n$. Table I shows the sufficient number of iterations and run times that are needed for the algorithms to successfully recover 80 percent of the original atoms. Table I shows that the 2D methods need fewer iterations and much less time to recover the dictionary. Moreover, the average time of each algorithms' iteration, which grows exponentially for 1D methods, is shown in Fig. 2.

[2]For SeDiL, we have used the codes available at http://www.gol.ei.tum.de/index.php.




(a) Recovery Percentage, $s = 7$, $L = 5000$ (b) Recovery Percentage, $s = 7$, $L = 10000$ (c) Recovery Percentage, $s = 15$, $L = 10000$

(d) RMSE, $s = 7$, $L = 5000$ (e) RMSE, $s = 7$, $L = 10000$ (f) RMSE, $s = 15$, $L = 10000$

Fig. 1. Successful recovery percentage and RMSE for $n = 10$ with different sparsity levels ($s$) and number of training signals ($L$) at SNR $= 30$ dB. SeDiL [17] is not included because it failed to recover the dictionary.

TABLE I
AVERAGE NUMBER OF ITERATIONS AND REQUIRED TIMES TO ACHIEVE 80 PERCENT RECOVERY(TIMES IN SECONDS, REPORTED BETWEEN BRACES). SPARSITY LEVEL $s = n$, AND $L = 1000n$.

| Signals size | $n = 10$ | $n = 15$ | $n = 20$ | $n = 25$ |
|---|---|---|---|---|
| 1D-MOD | 62(90) | 59(584) | 70(5110) | — |
| 1D-KSVD | 52(527) | 48(3339) | 65(18720) | — |
| 2D-MOD | 59(47) | 36(72) | 34(146) | 40(352) |
| 2D-CMOD | **24(20 )** | **23(49)** | **28(129)** | **25(235 )** |

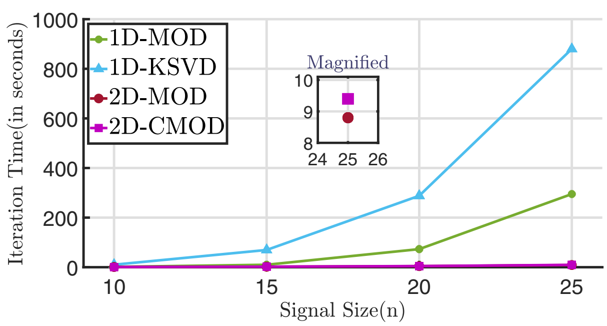



Fig. 2. Average time of each algorithms' iteration (in seconds). Sparsity level equals to $n$. The difference of computational times to recover 80 percent of the dictionary is more than the above difference because 2D algorithms need much fewer iterations, as shown in Table I.

### *B. Image Denoising*

Based on [21], forty thousand $12 \times 12$ patches are extracted from an image corrupted by additive white Gaussian noise with different standard deviations $\sigma_{noise} = 10, 20, 30, 50$. These signals are used as training signals for dictionary learning. In 1D methods, a dictionary of size $144 \times 576$ is learned, and in 2D case, two dictionaries of size $12 \times 24$ are learned. Therefore, the 1D and 2D representations have both 576 atoms, but the difference is in terms of memory; the memory of 1D methods is of order 82944 against 576 for 2D methods. The ODCT dictionary is used as an initial dictionary in both cases. 2D-OMP with stopping criteria $\|\mathbf{Y} - \mathbf{AXB}^T\|_F^2 \leq (1.15\sigma_{noise})^2$ is used for denoising all the overlapping image-patches. The same parameters as [21] are chosen, and all the algorithms are run for 30 iterations to have fair results. Peak Signal to-Noise-Ratio (PSNR) between the original and recovered images is used as a performance criterion, and the results are shown in Table II. Also, the total running times of algorithms (in seconds) are reported. Timings include dictionary training.

As we expected, the final quality of our proposed methods is a little bit less than the KSVD, because of the separable structure assumed for the dictionary. However, the reconstruction time is much less than KSVD, and the quality is better than fixed dictionaries. Moreover, the simulations illustrate that the approximation (10) is very efficient since the PSNR is very close to the PSNR obtained with 2D-MOD. More importantly, note that the difference of computational time grows by growing the patch size, as seen in Fig. 2.

## IV. CONCLUSION

In this paper, we introduced two algorithms for 2D dictionary learning. The first one was based on MOD. Then, a new objective function was achieved by using first order series expansion of $\mathbf{AXB}^T$, and 2D-CMOD algorithm was proposed to minimize it. Experimental results showed that the convergence rate of 2D-CMOD is much faster than 2D-MOD. Moreover, it was seen that the Kronecker dictionary is recovered better than 1D methods with much fewer training signals and much less computational cost. Also, the proposed methods were evaluated in image denoising application. Applying these methods to image compression and extending them to higher dimensions are as future works.

TABLE II
PSNR IN dB, AND TOTAL RUNNING TIMES IN SECONDS FOR DENOISING TWO IMAGES IN LEARNED DICTIONARIES. PARAMETERS ARE SET THE SAME AS [21], AND ALL THE ALGORITHMS ARE RUN FOR 30 ITERATIONS. THE VALUES ARE REPORTED BY AVERAGING OVER FIVE EXPERIMENTS.

| Images | boat | | | | house | | | | Total Time |
|---|---|---|---|---|---|---|---|---|---|
| $\sigma_{noise}$ (PSNR(dB)) | 10(28.12) | 20(22.12) | 30(18.61) | 50(14.13) | 10(28.18) | 20(22.12) | 30(18.60) | 50(14.15) | |
| ODCT (Not Trained) | 33.24 | 29.47 | 27.33 | 24.92 | 35.19 | 31.86 | 29.43 | 27.13 | 13 |
| 2D-MOD | 33.33 | 29.70 | 27.60 | 25.19 | 35.22 | 32.16 | 29.76 | 27.47 | 524 |
| 2D-CMOD | 33.26 | 29.59 | 27.57 | 25.17 | 35.03 | 31.98 | 29.69 | 27.44 | 636 |
| SeDiL | 31.14 | 27.20 | 25.20 | 23.47 | 32.91 | 29.00 | 26.39 | 24.32 | 573 |
| KSVD | 33.47 | 30.04 | 27.93 | 25.47 | 35.98 | 33.36 | 31.33 | 28.60 | 3130 |

# REFERENCES


[1] M. Elad, *Sparse and Redundant Representations: From Theory to Applications in Signal and Image Processing*, 1st ed. Springer Publishing Company, Incorporated, 2010.

[2] B. Dumitrescu and P. Irofti, *Dictionary Learning Algorithms and Applications*. Springer, 2018.

[3] S. Mallat and Z. Zhang, "Matching pursuits with time-frequency dictionaries," *IEEE Trans. on Signal Proc.*, vol. 41, no. 12, pp. 3397–3415, 1993. [Online]. Available: citeseer.ist.psu.edu/mallat93matching.html

[4] Y. Pati, R. Rezaiifar, and P. Krishnaprasad, "Orthogonal matching pursuit: Recursive function approximation with applications to wavelet decomposition," vol. 1, 12 1993, pp. 40 – 44 vol.1.

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

[6] H. Mohimani, M. Babaie-Zadeh, and C. Jutten, "A fast approach for overcomplete sparse decomposition based on smoothed $\ell^0$ norm," *IEEE Trans. on Signal Processing*, vol. 57, pp. 289–301, 2009.

[7] I. Daubechies, R. DeVore, M. Fornasier, and C. S. Güntürk, "Iteratively re-weighted least squares minimization for sparse recovery," *Communications on Pure and Applied Mathematics*, vol. 63, no. 1, pp. 1–38, Jan. 2010.

[8] M. Aharon, M. Elad, A. Bruckstein *et al.*, "K-svd: An algorithm for designing overcomplete dictionaries for sparse representation," *IEEE Transactions on signal processing*, vol. 54, no. 11, p. 4311, 2006.

[9] K. Engan, S. O. Aase, and J. H. Husoy, "Method of optimal directions for frame design," in *Acoustics, Speech, and Signal Processing, 1999. Proceedings., 1999 IEEE International Conference on*, vol. 5. IEEE, 1999, pp. 2443–2446.

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

[11] J. Parsa, M. Sadeghi, M. Babaie-Zadeh, and C. Jutten, "Low mutual and average coherence dictionary learning using convex approximation," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 3417–3421.

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

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

[14] J. Parsa, M. Sadeghi, M. Babaie-Zadeh, and C. Jutten, "Joint low mutual and average coherence dictionary learning," in *2018 26th European Signal Processing Conference (EUSIPCO)*. IEEE, 2018, pp. 1725–1729.

[15] R. C. Gonzalez and R. E. Woods, *Digital Image Processing (3rd Edition)*. Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 2006.

[16] A. Ghaffari, M. Babaie-Zadeh, and C. Jutten, "Sparse decomposition of two dimensional signals," in *Acoustics, Speech and Signal Processing, 2009. ICASSP 2009. IEEE International Conference on*. IEEE, 2009, pp. 3157–3160.

[17] S. Hawe, M. Seibert, and M. Kleinsteuber, "Separable dictionary learning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2013, pp. 438–445.

[18] K. B. Petersen, M. S. Pedersen, J. Larsen, K. Strimmer, L. Christiansen, K. Hansen, L. He, L. Thibaut, M. B. ao, S. Hattinger, V. Sima, and W. The, "The matrix cookbook," Tech. Rep., 2008.

[19] Y. Fang, J. Wu, and B. Huang, "2D sparse signal recovery via 2D orthogonal matching pursuit," *Science China Information Sciences*, vol. 55, no. 4, pp. 889–897, 2012.

[20] J. Parsa, M. Sadeghi, M. Babaie-Zadeh, and C. Jutten, "A new algorithm for dictionary learning based on convex approximation," in *2019 27th European Signal Processing Conference (EUSIPCO)*. IEEE, 2019, pp. 1–5.

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