LEARNING PHASE-INVARIANT DICTIONARIES

Graeme Pope[1], *Céline Aubel*[1], *and Christoph Studer*[2]

[1]Dept. of Inf. Technology & Electrical Eng., ETH Zurich, Switzerland, {gpope, aubelc}@nari.ee.ethz.ch
[2]Dept. of Electrical & Computer Eng., Rice University, Houston, TX, USA, studer@rice.edu

ABSTRACT

In this paper, we present a novel algorithm to learn phase-invariant dictionaries, which can be used to efficiently approximate a variety of signals, such as audio signals or images. Our approach relies on finding a small number of generating atoms that can be used—along with their phase-shifts—to sparsely approximate a given signal. Our method is inspired by the K-SVD algorithm, but imposes an extra constraint that the dictionaries we learn are phase-invariant. We show that the learned dictionaries achieve competitive approximation performance compared to that of state-of-the-art methods for audio signals and images, while substantially reducing the storage requirements and computational complexity.

***Index Terms*—** Dictionary learning, K-SVD, sparse approximation, phase-invariant and shift-invariant dictionaries.

1. INTRODUCTION

Dictionary learning (DL) [1–3] was shown to be very effective for sparsity-based denoising [3–5], audio and music analysis [6], super-resolution [7, 8], and inpainting [9], by relying on sparse representations of the given signals using a specifically trained dictionary (overcomplete and standardized matrix). In many applications, such as audio and image processing, the entire signal is decomposed into small patches (or segments) by exploiting the fact that these patches tend to be self-similar (see, e.g., [10]) and admit a sparse representation using a learned dictionary. In many cases, when computing a sparse representation of a signal, the same dictionary elements (atoms) are used to represent different patches, but which may have entirely different phase shifts.

1.1. Phase-invariant dictionaries

Phase-invariant (PI) dictionaries are an effective way of representing signal patches, using the same atoms, but with different phase shifts. Specifically, a PI dictionary $\mathbf{D}^{\circledcirc}$ consists of all the circularly shifted versions of a (small) set of generating elements, where each circular shift corresponds to a phase change of the signal. Let $\mathbf{c}$ be a patch or segment of a signal, for instance of an audio signal, $\mathbf{c} \in \mathbb{R}^{m_1}$, and for images, $\mathbf{c} \in \mathbb{R}^{m_1 \times m_2}$. More generally, assume that we have a patch $\mathbf{c} \in \mathbb{R}^m$ with $m = m_1 \times \cdots \times m_d$. Then we can circularly shift (or rotate) $\mathbf{c}$ in all dimensions, thereby generating $m-1$ circularly shifted-versions of $\mathbf{c}$. We say that $\mathbf{c}$ generates $\mathbf{C} \in \mathbb{R}^{m \times m}$ if each column of $\mathbf{C}$ is the vectorized version of a differently shifted version of $\mathbf{c}$. In a slight abuse of notation, we will say that the matrix $\mathbf{C}$ is circulant, which is, strictly speaking, only the case for $d = 1$. Then, given a set of unit ℓ_2-norm vectors $\mathbf{c}_\ell$, $\ell = 1, \ldots, p$, that generate the circulant matrices $\mathbf{C}_\ell$, $\mathbf{D}^{\circledcirc} = [\mathbf{C}_1, \ldots, \mathbf{C}_p]$ is a PI dictionary. And we call the vectors $\mathbf{c}_\ell$ the *generating atoms* of $\mathbf{D}^{\circledcirc}$.

Although one could learn a PI dictionary by first using standard DL methods [1–3] and then constructing a PI dictionary by incorporating all possible phase shifts of the learned dictionary elements, we will show that incorporating this particular structure in the DL procedure itself results in a number of advantages. Concretely, such phase-invariant dictionaries achieve competitive approximation performance compared to dictionaries obtained through state-of-the-art methods, like K-SVD [3], while significantly reducing the storage requirements, as well as the computational complexity of both dictionary learning and signal representation/reconstruction.

As an example of the advantage of PI dictionaries, consider a periodic audio signal $a_1, a_2, a_3, a_4, a_1, a_2, a_3, a_4, \ldots$, which we want to approximate using a single dictionary element (atom). In this case, the atom $\mathbf{c} = (a_1, a_2, a_3, a_4)^{\mathsf{T}}$ would be a suitable choice. However, if one takes a chunk of this periodic sequence starting at an arbitrary sample index, e.g., a_3, a_4, a_1, a_2, it cannot be represented efficiently by $\mathbf{c}$. But by allowing the use of circularly shifted versions of $\mathbf{c}$, any patch can be efficiently represented.

1.2. Contributions and outline

In what follows, we present a novel DL algorithm (Sec. 3), which is able to learn phase-invariant dictionaries $\mathbf{D}^{\circledcirc}$, which consist of the concatenation of p "circulant" matrices, i.e., $\mathbf{D}^{\circledcirc} = [\mathbf{C}_1, \ldots, \mathbf{C}_p]$. The proposed algorithm is inspired by the K-SVD algorithm [3] (Sec. 2), and replaces the singular-value decomposition step with a dictionary update procedure relying on the fast Fourier transform, which reduces the com-

This work was supported in part by the Swiss National Science Foundation under Grant PA00P2-134155. The authors would like to thank H. Bölcskei for his support and encouragement.

plexity of the proposed DL algorithm. We finally demonstrate the advantages (in terms of storage requirements, learning complexity, and representation accuracy) of PI dictionaries and our DL algorithm over existing methods (Sec. 4).

1.3. Relevant prior work

There exists a large body of related results on learning of *shift-invariant dictionaries* [11–18]. Concretely, the publications in [11–14] consider a single "long" signal as training data and then attempt to find "small" patches to approximate the entire signal. In these papers, the shifts are generally *not allowed* to be circulant, unlike in our setting, where we explicitly consider circularly shifted dictionary elements. Consequently, these methods are ill suited to handle the case with dictionaries consisting of a concatenation of circulant matrices.

Circulant shifts are considered in [15], but their algorithm only allows a single phase-shift to be used per generating element, i.e., for each $\mathbf{C}_\ell$, at most one column can contribute to the sparse representation. Consequently, they need to learn much larger dictionaries than required by our generalized approach. In [16] they take a slightly different route and aim to learn a dictionary that is, in general, not circulant, but where the training signals contain all possible shifts of the training patches. To deal with the potentially very large training sets, a subspace clustering method is used for DL. The method described in [11] is based on the earlier work of [1] and uses a shift-invariant generative model that describes the signal as a sparse linear combination of atoms and of all their shifted versions. The parameters of this model are then learned using maximum likelihood estimation and a stochastic gradient descent method to update the dictionary, as in [17, 18], where, in contrast, our update method is given in closed form. Seemingly similar results have been developed in [19]. There, however, a single circulant matrix is learned, that—when subsampled—is incoherent to a sparsity basis; this problem differs from our goal as it amounts to designing good measurement matrices for compressive sensing [20, 21].

2. DICTIONARY LEARNING VIA K-SVD

DL algorithms attempt to find a matrix $\mathbf{D} = [\mathbf{d}_1, \dots, \mathbf{d}_n] \in \mathbb{R}^{m\times n}$ with unit ℓ_2-norm columns that can be used to sparsely represent a collection of training data $\mathbf{Y} = [\mathbf{y}_1, \dots, \mathbf{y}_T] \in \mathbb{R}^{m\times T}$. That is, for each $j = 1, \dots, T$, we want to find a vector $\mathbf{x}_j$ that is sparse and satisfies $\mathbf{y}_j \approx \mathbf{D}\mathbf{x}_j$. The problem is then, given $\mathbf{Y} \in \mathbb{R}^{m\times T}$, learn a dictionary $\mathbf{D} \in \mathbb{R}^{m\times n}$ and a sparse matrix $\mathbf{X} \in \mathbb{R}^{n\times T}$ so that $\|\mathbf{Y} - \mathbf{D}\mathbf{X}\|_\mathrm{F}$ is small. More formally, we seek to solve the following problem:

$$\text{(DL)} \quad \begin{cases} \underset{\mathbf{D},\mathbf{X}}{\text{minimize}} & \|\mathbf{Y} - \mathbf{D}\mathbf{X}\|_\mathrm{F}^2 \\ \text{subject to} & \|\mathbf{x}_j\|_0 \leqslant S, \forall j, \ \|\mathbf{d}_\ell\|_2 = 1, \forall \ell, \end{cases}$$

where $\|\mathbf{x}_j\|_0$ designates the number of non-zero entries in $\mathbf{x}_j$.

The K-SVD algorithm [3] is designed to efficiently compute an approximate to the (DL) problem. After initializing $\mathbf{D}$ with random or well-defined data, the K-SVD algorithm performs the following two steps at each iteration:

1) Sparse update: In this step the dictionary $\mathbf{D}$ is held constant and the columns of $\mathbf{X}$ are updated. Since we can rewrite the objective of (DL) as

$$\|\mathbf{Y} - \mathbf{D}\mathbf{X}\|_\mathrm{F}^2 = \textstyle\sum_{j=1}^{T} \|\mathbf{y}_j - \mathbf{D}\mathbf{x}_j\|_2^2\,,$$

each column $\mathbf{x}_j$ of $\mathbf{X}$ can be updated individually by solving the *primal sparse approximation problem* (PSAP) [22]

$$\text{(PSAP)} \quad \text{minimize } \|\mathbf{y}_j - \mathbf{D}\mathbf{x}_j\|_2 \ \text{ subject to } \|\mathbf{x}_j\|_0 \leqslant S,$$

which can be approximated using orthogonal matching pursuit (OMP) [22] or other sparse signal recovery methods.

2) Dictionary update: After updating the matrix $\mathbf{X}$, the dictionary $\mathbf{D}$ and the non-zero entries of $\mathbf{X}$ are jointly updated. To do this, the K-SVD algorithm isolates a single column of $\mathbf{D}$ as follows:

$$\|\mathbf{Y}-\mathbf{D}\mathbf{X}\|_\mathrm{F}^2 = \|\mathbf{Y}-\textstyle\sum_{\ell=1}^{n} \mathbf{d}_\ell[\mathbf{X}]_\ell\|_\mathrm{F}^2 = \|\mathbf{E}_k-\mathbf{d}_k[\mathbf{X}]_k\|_\mathrm{F}^2\,,$$

where $\mathbf{E}_k = \mathbf{Y} - \sum_{\ell\neq k} \mathbf{d}_\ell[\mathbf{X}]_\ell$ and $[\mathbf{X}]_\ell$ denotes the ℓth row of $\mathbf{X}$. Now let Ω_k be the support of $[\mathbf{X}]_k$. To update $\mathbf{d}_k$ and $[\mathbf{X}]_k$, one sets $\mathbf{E}_k^R$ to be the matrix obtained by taking only the columns of $\mathbf{E}_k$ with indices in Ω_k, $[\mathbf{X}]_k^R$ to be the row vector $[\mathbf{X}]_k$ restricted to the entries with indices in Ω_k. Then minimize $\left\|\mathbf{E}_k^R - \mathbf{d}_k[\mathbf{X}]_k^R\right\|_\mathrm{F}$ with respect to $\mathbf{d}_k$ and $[\mathbf{X}]_k^R$, which amounts to finding the best rank-one approximation to $\mathbf{E}_k^R$, given by its singular value decomposition. Thus, $\mathbf{d}_k$ is set to the dominant left singular vector and $[\mathbf{X}]_k^R$ to the dominant singular value times the dominant right singular vector.

3. LEARNING PHASE-INVARIANT DICTIONARIES

A straightforward way of using K-SVD to generate phase-invariant dictionaries is to first learn a dictionary $\mathbf{D} \in \mathbb{R}^{m\times p}$ and then, to generate a new dictionary $\mathbf{D}_\mathsf{C} \in \mathbb{R}^{m\times mp}$ consisting of all the atoms of $\mathbf{D}$ and their circular shifts[1]. To achieve substantially better performance than the straightforward method outlined above (see Sec. 4), our aim is to *directly* learn a phase-invariant dictionary $\mathbf{D}^\circledcirc$ from $\mathbf{Y}$.

3.1. Algorithm

Adapting the *sparse update* step of the K-SVD algorithm using OMP is straightforward, as there is nothing to change. The *dictionary update* step, however, must be designed from scratch. Concretely, a column-wise update is no longer possible as the columns of $\mathbf{D}^\circledcirc$ are no longer independent. Hence, one needs to perform a block update step by solving

$$\underset{\mathbf{C}_k \text{ circulant}}{\text{minimize}} \ \|\mathbf{E}_k - \mathbf{C}_k\mathbf{X}[k]\|_\mathrm{F}^2\,, \tag{1}$$

[1] We refer to this approach as "circularizing $\mathbf{D}$."

where $\mathbf{E}_k = \mathbf{Y} - \sum_{\ell \neq k} \mathbf{C}_\ell \mathbf{X}[\ell]$ and $\mathbf{X}[k]$ is the submatrix of $\mathbf{X}$ consisting of the rows $m(k-1)+1$ to mk of $\mathbf{X}$. Unfortunately, there is no obvious way of solving (1) directly.

The key observation for solving the problem in (1) is that we can diagonalize the circulant submatrices $\mathbf{C}_k$ using the discrete Fourier transform (DFT) and thus, isolate the individual generating elements $\mathbf{c}_k$. Let a column of $\mathbf{C}_k$ be the vectorized form of a $m_1 \times \cdots \times m_d$ patch in d-dimensions so that $m = m_1 \cdots m_d$ and let $\mathbf{F}_n$ be the $n \times n$ DFT matrix. Then, the matrix $\mathbf{F} = \mathbf{F}_{m_1} \otimes \cdots \otimes \mathbf{F}_{m_d}$ diagonalizes each $\mathbf{C}_\ell$, that is $\mathbf{\Lambda}_\ell = \mathbf{F}^* \mathbf{C}_\ell \mathbf{F}$ is a diagonal matrix.[2] Hence, the dictionary to be learned takes the form $\mathbf{D}^{\circledcirc} = \mathbf{F}^* \mathbf{\Lambda}(\mathbf{I}_m \otimes \mathbf{F})$ with the matrix $\mathbf{\Lambda} = [\mathbf{\Lambda}_1, \ldots, \mathbf{\Lambda}_p]$ having a single non-zero entry per column. Thus, the problem we want to solve is

$$\begin{aligned} &\underset{\mathbf{X},\mathbf{D}^{\circledcirc}}{\text{minimize}} \ \left\|\mathbf{Y} - \mathbf{D}^{\circledcirc}\mathbf{X}\right\|_{\mathrm{F}} \\ &\text{subject to } \mathbf{D}^{\circledcirc} = [\mathbf{F}^*\mathbf{\Lambda}_1\mathbf{F}, \ldots, \mathbf{F}^*\mathbf{\Lambda}_p\mathbf{F}] \text{ with } \mathbf{\Lambda}_k \text{ diagonal}, \\ &\qquad \|\mathbf{x}_j\|_0 \leqslant S, \ \forall j, \ \left\|\mathbf{d}_\ell^{\circledcirc}\right\|_2 = 1, \ \forall \ell. \end{aligned}$$

We can rewrite this problem as follows:

$$(\text{PI-DL}) \begin{cases} \underset{\mathbf{X},\mathbf{\Lambda}_1,\ldots,\mathbf{\Lambda}_p}{\text{minimize}} & \|\mathbf{Y} - \mathbf{F}^*[\mathbf{\Lambda}_1, \ldots, \mathbf{\Lambda}_p](\mathbf{I}_m \otimes \mathbf{F})\mathbf{X}\|_{\mathrm{F}} \\ \text{subject to} & \mathbf{\Lambda}_k \text{ is diagonal}, \ k = 1, \ldots, p \\ & \|\mathbf{x}_j\|_0 \leqslant S, \ \forall j, \ \left\|\mathbf{d}_\ell^{\circledcirc}\right\|_2 = 1, \forall \ell, \\ & \mathbf{D}^{\circledcirc} = \mathbf{F}^*[\mathbf{\Lambda}_1, \ldots, \mathbf{\Lambda}_p](\mathbf{I}_m \otimes \mathbf{F}). \end{cases}$$

In order to solve (PI-DL), we follow the alternating optimization approach used in the classical K-SVD algorithm [3]. In the first stage, we update the matrix $\mathbf{X}$ and in the second stage we update the dictionary $\mathbf{D}^{\circledcirc}$.

1) Sparse update: To update the jth column of $\mathbf{X}$, find the best S-sparse approximation to $\mathbf{y}_j$ using $\mathbf{F}\mathbf{\Lambda}(\mathbf{I}_m \otimes \mathbf{F})$ by solving the following PSAP [22]

$$\underset{\mathbf{x}}{\text{minimize}} \ \|\mathbf{y}_j - \mathbf{F}\mathbf{\Lambda}(\mathbf{I}_m \otimes \mathbf{F})\mathbf{x}\|_2 \ \text{ subject to } \ \|\mathbf{x}\|_0 \leqslant S,$$

which can be approximated using OMP (cf. Sec. 2).

2) Dictionary update: To update the diagonal matrices $\mathbf{\Lambda}_k$ in PI-DL, we again isolate each column (which contains only a single non-zero entry), and then find the value which minimizes the objective function.

More specifically, let $\lambda_{\ell,k}$ be the ℓth diagonal element of $\mathbf{\Lambda}_k$, and let $\mathbf{X}[k] \in \mathbb{R}^{m \times T}$ be the submatrix of $\mathbf{X}$ consisting of the rows $m(k-1)+1$ to mk of $\mathbf{X}$. Since the matrix $\mathbf{F}$ is unitary, we have that

$$\begin{aligned} &\|\mathbf{Y} - \mathbf{F}^*\mathbf{\Lambda}(\mathbf{I}_m \otimes \mathbf{F})\mathbf{X}\|_{\mathrm{F}}^2 = \|\mathbf{F}\mathbf{Y} - \mathbf{\Lambda}(\mathbf{I}_m \otimes \mathbf{F})\mathbf{X}\|_{\mathrm{F}}^2 \\ &= \|\mathbf{F}\mathbf{Y} - \textstyle\sum_{k=1}^p \mathbf{\Lambda}_k\mathbf{F}\mathbf{X}[k]\|_{\mathrm{F}}^2 = \|\mathbf{Z}_\ell - \mathbf{\Lambda}_\ell\mathbf{F}\mathbf{X}[\ell]\|_{\mathrm{F}}^2, \quad (2) \end{aligned}$$

where $\mathbf{Z}_\ell = \mathbf{F}\mathbf{Y} - \sum_{k \neq \ell} \mathbf{\Lambda}_k \mathbf{F}\mathbf{X}[k]$. Decomposing (2) row-wise then yields

$$\|\mathbf{Y} - \mathbf{F}^*\mathbf{\Lambda}(\mathbf{I}_m \otimes \mathbf{F})\mathbf{X}\|_{\mathrm{F}}^2 = \textstyle\sum_{j=1}^m \|[\mathbf{Z}_\ell]_j - \lambda_{j,\ell}\,[\mathbf{F}\mathbf{X}[\ell]]_j\|_2^2,$$

[2] The operator $\otimes$ corresponds to the Kronecker product.

Table 1: Properties of the smallest dictionary required to get an SNR exceeding 20 dB. DoF is the degrees of freedom which is the number of entries required to specify the dictionary, Gen. is the number of generating atoms (if applicable), and L.T. is the time required to learn the dictionary.

	Atoms	Gen.	DoF	L.T.	SNR
K-SVD	129	–	8,192	133s	20.8dB
K-SVD ⊚	1,025	16	1,024	92s	20.0dB
PI-DL ⊚	321	5	320	88s	20.4dB
SI-DL	193	–	12,288	304s	20.6dB

where $[\mathbf{F}\mathbf{X}[\ell]]_j$ is the jth row of $\mathbf{F}\mathbf{X}[\ell]$. We can now minimize the above expression by differentiating it with respect to $\lambda_{i,\ell}$ and setting the result equal to zero. Thus, for each i, we compute the following expression:

$$\lambda_{i,\ell} = [\mathbf{Z}_\ell]_i\,[\mathbf{F}\mathbf{X}[\ell]]_i^* \times \|[\mathbf{F}\mathbf{X}[\ell]]_i\|_2^{-2}.$$

After calculating $\mathbf{D}^{\circledcirc}$ from the updated $\mathbf{\Lambda}_k$ matrices, we normalize each column of $\mathbf{D}^{\circledcirc}$ by setting $\mathbf{d}_i = \mathbf{d}_i / \|\mathbf{d}_i\|_2$ and then multiplying the ith row of $\mathbf{X}$ by $\|\mathbf{d}_i\|_2$ as put forward in [4]. Note that for signals with non-zero mean, it is often useful to append an atom of all ones to the resulting dictionary followed by normalizing it to unit ℓ_2-norm.

3.2. Storage and implementation speed

A significant feature of PI dictionaries is that they have low storage requirements, as only the p generating atoms need to be stored. This is of particular importance when storage and memory access comes with a premium, as, e.g., in hardware implementations [23]. Alternatively, for the same storage capacity as unstructured dictionaries, the reconstruction performance is improved as the circularized dictionary contains more columns while only storing the generating elements. In addition, for streaming applications, the incoming data is typically divided into small patches, which can then be processed independently. Such small patches are more likely to contain phase-shifted versions of the dictionary atoms.

Another advantage of PI dictionaries is that matrix-vector multiplications can be efficiently implemented using the fast Fourier transform (FFT). For an $m \times n$ PI dictionary $\mathbf{D}^{\circledcirc}$ with $p = n/m$ generating atoms, the matrix-vector multiplication $\mathbf{D}^{\circledcirc}\mathbf{x}$ can be performed with $\mathcal{O}(n \log m)$ operations, whereas for a non-PI dictionary of the same size it requires $\mathcal{O}(mn)$ operations. This complexity gain is of particular importance for algorithms such as BP [21] or AMP [23], which make heavy use of this matrix-vector multiplication step.

4. RESULTS AND APPLICATION EXAMPLE

4.1. Synthetic results

We now assess the performance and complexity of PI-DL using synthetic experiments, comparing to K-SVD [3] and shift-

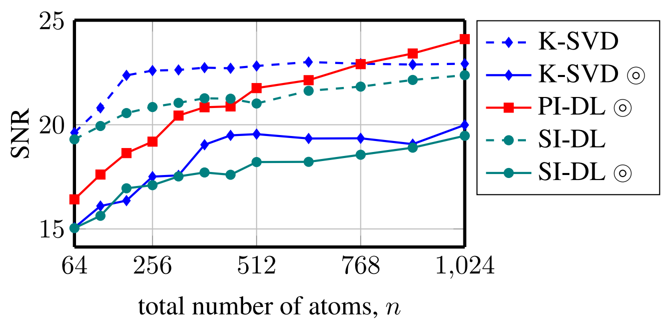


Fig. 1: 8-sparse approximation error for audio data. ⊚ denotes that the dictionary is phase-invariant.

invariant DL, (SI-DL), [14].[3] Since the implementation of SI-DL[4] was only available for audio signals, we consider an audio signal approximation application. Thus we learn the following $m \times n$ dictionaries, with $m = 64$ and $p = (n-1)/m$, from a piece of music[5] with

(i) n atoms using K-SVD [3] and SI-DL [14]
(ii) p generating atoms using K-SVD, PI-DL, or SI-DL.

If we learn p generating vectors, we circularize the dictionary, i.e., we include all the shifted versions of the generating atoms. We include a column of ones in the learned dictionary and try to learn the best dictionary whilst using an 8-sparse representation of the training data. We then approximate a random segment from the remainder of the movement using these dictionaries to get the SNR, defined by $20 \log_{10}(\|\mathbf{y}\|_2 \,/\, \|\mathbf{y} - \hat{\mathbf{y}}\|_2)$, where $\mathbf{y}$ is the original data and $\hat{\mathbf{y}}$ the 8-sparse approximation with the learned dictionary.[6]

From Fig. 1, we see that the circularized PI-DL approach gives the best SNR out of all the circularized methods. K-SVD and SI-DL can perform better than PI-DL, but this is unsurprising since the resulting dictionary has 64× more degrees of freedom than the PI-DL dictionary. For large values of n, PI-DL outperforms both SI-DL and K-SVD algorithm; this is mainly because SI-DL and K-SVD are overfitting their dictionaries to the training data.

In Tab. 1 we compare the times taken to generate the smallest dictionary that gives an SNR of at least 20 dB.[7] Although PI-DL ⊚ requires a large dictionary, it is in fact specified by the smallest number of parameters, as we only need to store the 5 generating vectors. It is also considerably faster than the other methods. Thus, PI-DL learns a smaller dictionary, faster than the other approaches, and results in a better sparse representation of the data.

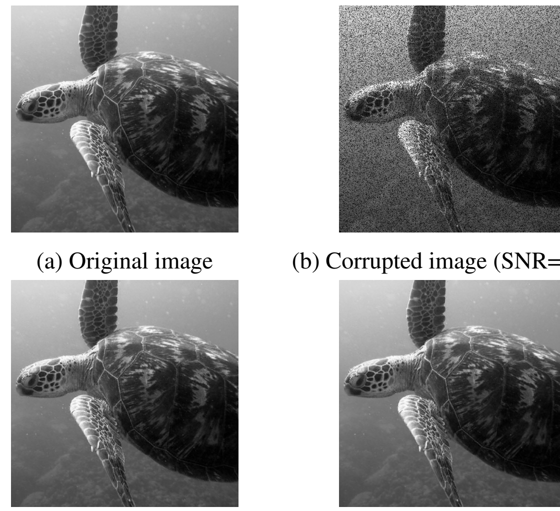
[3] We also implemented the MoTIF algorithm [12], but SI-DL performed substantially better and hence, for brevity, we omit the MoTIF results.

[4] Available at: `http://code.soundsoftware.ac.uk/projects/siksvd/repository`

[5] Precisely, we use a random patch selection from the first 10 s of Mozart's K452 Largo - Allegro moderato, as training data.

[6] All simulations were performed on a i7 2.2 GHz laptop computer with 8 GB RAM and running Matlab R2012a.

[7] Note that for SI-DL ⊚, we were not able to achieve this error rate, hence the results are not shown.



(a) Original image (b) Corrupted image (SNR=7 dB)

(c) PI-DL (SNR=28.0 dB) (d) K-SVD (SNR=28.2 dB)

Fig. 2: Reconstructed images using PI-DL and K-SVD. 12 generating vectors were learned for PI-DL ($n = 769$), taking 105s and for K-SVD, $n = 257$, taking 739s to learn.

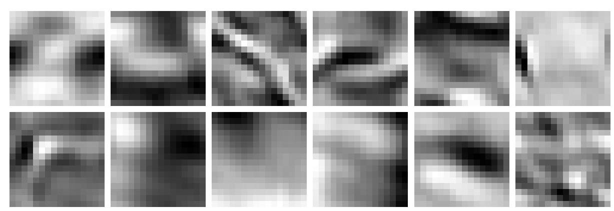


Fig. 3: The $p = 12$ generating atoms learned using PI-DL.

4.2. Application example

We now compare PI-DL to K-SVD for a simple inpainting example. Here, we use a picture of a turtle as training data to generate a dictionary and attempt to reconstruct a different turtle picture that is missing 30% of its pixels. In Fig. 2 we show the corrupted image and the reconstructed images. In Fig. 3 we show the 12 generating atoms learned by PI-DL. Similar to the audio data, PI-DL learns a dictionary 7× faster and specified by 20× fewer parameters, for the same reconstruction performance. Hence, PI-DL is much more efficient from a complexity and storage requirement perspective.

5. CONCLUSION

We have shown that phase-invariant (PI) dictionaries can give the same or better reconstruction performance as regular (unstructured) dictionaries, but require much less time to generate and need much less storage space. This feature is particularly important for hardware implementations, where memory access and storage comes at a premium. PI dictionaries are suitable for many applications including audio/image denoising and inpainting. In addition, their combination of low storage requirements and fast implementation renders them particularly attractive for the use in hardware implementations.

6. REFERENCES

[1] B. A. Olshausen and D. J. Field, "Emergence of simple-cell receptive field properties by learning a sparse code for natural images," *Nature*, vol. 13, pp. 607–609, June 1996.

[2] ——, "Sparse coding with an overcomplete basis set: A strategy employed by V1," *Vision Research*, vol. 37, pp. 3311–3325, Dec. 1997.

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

[4] C. Studer and R. G. Baraniuk, "Dictionary learning from sparsely corrupted or compressed signals," in *Proc. of IEEE Int. Conf. Acoustics, Speech, and Sig. Process. (ICASSP)*, Kyoto, Japan, Mar. 2012, pp. 3341–3344.

[5] T. Faktor, Y. C. Eldar, and M. Elad, "Denoising of image patches via sparse representations with learned statistical dependencies," in *Proc. of IEEE Int. Conf. Acoustics, Speech, and Sig. Process. (ICASSP)*, Prague, Czech Republic, May 2011, pp. 5820–4823.

[6] 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. of the IEEE*, vol. 98, no. 6, pp. 995–1005, June 2010.

[7] 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, Dec. 2006.

[8] J. Yang, J. Wright, T. Huang, and Y. Ma, "Image super-resolution via sparse representation," *IEEE Trans. Image Process.*, vol. 19, no. 11, pp. 2861–2872, Nov. 2010.

[9] J. Mairal, F. Bach, J. Ponce, and G. Sapiro, "Online dictionary learning for sparse coding," in *Proc. of the 26th Ann. Int. Conf. on Mach. Learning*, Montreal, Canada, June 2009, pp. 689–696.

[10] M. Aharon and M. Elad, "Sparse and redundant modeling of image content using an image-signature-dictionary," *SIAM J. Imaging Sci.*, vol. 1, pp. 228–247, July 2008.

[11] T. Blumensath and M. E. Davies, "Sparse and shift-invariant representations of music," *IEEE Trans. on Audio, Speech, and Language Process.*, vol. 14, no. 1, pp. 50–57, Jan. 2006.

[12] P. Jost, P. Vandergheynst, S. Lesage, and R. Gribonval, "MoTIF: An efficient algorithm for learning translation invariant dictionaries," in *Proc. of IEEE Int. Conf. Acoustics, Speech, and Sig. Process. (ICASSP)*, Toulouse, France, May 2006, pp. 857–860.

[13] S. Lesage, "Apprentissage de dictionnaires structurés pour la modélisation parcimonieuse de signaux multicanaux," Ph.D. dissertation, Université de Rennes 1, 2007.

[14] B. Mailhé, S. Lesage, R. Gribonval, F. Bimbot, and P. Vandergheynst, "Shift-invariant dictionary learning for sparse representations: extending K-SVD," in *European Sig. Proc. Conf. (EUSIPCO)*, Lausanne, Switzerland, Aug. 2008.

[15] J. J. Thiagarajan, K. N. Ramamurthy, and A. Spanias, "Shift-invariant sparse representation of images using learned dictionaries," in *IEEE Mach. Learning Sig. Proc.*, Oct. 2008, pp. 145–150.

[16] B. V. Gowreesunker and A. H. Tewfik, "A shift tolerant dictionary training method," in *Signal Processing with Adaptive Sparse Structured Representations (SPARS)*, Saint Malo, France, Apr. 2009.

[17] M. Nakashizuka, H. Nishiura, and Y. Iiguni, "Sparse image representations with shift-invariant tree-structured dictionaries," in *IEEE Int. Conf. Image. Proc.*, Cairo, Egypt, Nov. 2009, pp. 2145–2148.

[18] Q. Barthélemy, A. Larue, A. Mayoue, D. Mercier, and J. I. Mars, "Shift & 2D rotation invariant sparse coding for multivariate signals," *IEEE Trans. Sig. Proc.*, vol. 60, no. 4, pp. 1597–1611, Apr. 2012.

[19] Y. Xu, W. Yin, and S. Osher, "Learning a circulant sensing kernel," CAAM, Rice University, Tech. Rep., Jan. 2012. [Online]. Available: http://www.caam.rice.edu/~wy1/paperfiles/Rice_CAAM_TR12-05_Learn_Circ_Mtx.PDF

[20] E. J. Candès, J. Romberg, and T. T. Tao, "Stable signal recovery from incomplete and inaccurate measurements," *Comm. Pure and Applied Math.*, vol. 59, no. 8, pp. 1207–1223, Mar. 2006.

[21] D. L. Donoho, "Compressed sensing," *IEEE Trans. Inf. Theory*, vol. 52, no. 4, pp. 1289–1306, Apr. 2006.

[22] J. A. Tropp, "Greed is good: Algorithmic results for sparse approximation," *IEEE Trans. Inf. Theory*, vol. 50, no. 10, pp. 2231–2242, Oct. 2004.

[23] P. Maechler, C. Studer, D. Bellasi, A. Maleki, A. Burg, N. Felber, H. Kaeslin, and R. G. Baraniuk, "VLSI implementation of approximate message passing for signal restoration and compressive sensing," *IEEE J. on Emerg. Sel. Topics in Circ. and Sys.*, vol. 2, no. 3, pp. 579–590, Oct. 2012.