Fast sparsity-based orthogonal dictionary learning for image restoration

Chenglong Bao[1], Jian-Feng Cai[2] and Hui Ji[1]
[1]Department of Mathematics, National University of Singapore, Singapore,119076
[2]Department of Mathematics, University of Iowa, Iowa City, IA, USA, 52242
{baochenglong,matjh}@nus.edu.sg, jianfeng-cai@uiowa.edu

Abstract

In recent years, how to learn a dictionary from input images for sparse modelling has been one very active topic in image processing and recognition. Most existing dictionary learning methods consider an over-complete dictionary, e.g. *the K-SVD method. Often they require solving some minimization problem that is very challenging in terms of computational feasibility and efficiency. However, if the correlations among dictionary atoms are not well constrained, the redundancy of the dictionary does not necessarily improve the performance of sparse coding. This paper proposed a fast orthogonal dictionary learning method for sparse image representation. With comparable performance on several image restoration tasks, the proposed method is much more computationally efficient than the over-complete dictionary based learning methods.*

1. Introduction

In recent years, sparse models for representing natural images have been an active research topic in computer vision and image processing community. It is now well established that the sparse image models are very powerful tools for many image restoration and recognition tasks. Sparse image model assumes that most local image patches can be well approximated by a sparse linear combination of basis elements, the so-called *atoms*. The collection of these elements is called a *dictionary*. A fundamental question is then how to find a dictionary under which an input image can be sparsely modelled. Earlier work on designing dictionary for sparse image modelling focuses on the design of fixed orthogonal dictionaries, *e.g.* local *discrete cosine transform* (DCT) [25], *wavelets* [7, 21]. These orthogonal dictionaries and their over-complete extensions (*e.g. tight frames* [8]) remain important tools in many image restoration tasks (*e.g.* [3, 5]) for their simplicity and efficiency. Recently, there have been great progresses on constructing dictionaries adaptive to the input image via some learning process (*e.g.* [12, 15, 19, 17]). The basic idea is to learn the dictionary adaptive to the target image so as to achieve better sparsity than the fixed ones. Most existing dictionary learning methods consider an over-complete dictionary and formulate the learning process as a minimization problem. Taking the popular K-SVD method [12] for example, the K-SVD method learns an over-complete dictionary from an input image via solving the following minimization model:

$$\min_{\boldsymbol{D},\{\boldsymbol{\alpha}_i\}} \sum_i \|\boldsymbol{g}_i - \boldsymbol{D}\boldsymbol{\alpha}_i\|_2^2 + \lambda_i \|\boldsymbol{\alpha}_i\|_0, \qquad (1)$$

where $\|\cdot\|_0$ is the sparsity measure defined as the number of non-zero entries in the input, $\{\boldsymbol{g}_i\} \subset \mathbb{R}^n$ is the collection of image patches after vecterization. $D = [\boldsymbol{d}_1, \ldots, \boldsymbol{d}_k] \subset \mathbb{R}^{n\times k}$ with $k > n$ is the unknown over-complete dictionary.

The problem (1) is indeed a very challenging non-convex minimization problem. The iteration scheme for solving (1) is proposed in [12] which alternatively iterates between two modules: sparse coding for $\{\boldsymbol{\alpha}_i\}$ and dictionary updating for $\boldsymbol{D}$. Both modules use some greedy approach which lacks rigorious theoretical treatment on its optimality and convergence. Moreover, they are very computational demanding. Since then, many methods have been proposed to either speed up the computation (*e.g.* [26]) or to modify the model (1) for better stability including replacing the non-convex ℓ_0 norm $\|\cdot\|_0$ by its convex relaxation ℓ_1 norm $\|\cdot\|_1$ (*e.g.* [1, 16]) or by the MC penalty [27]. However, the issues on computational efficiency and convergence are still not completely overcomed.

In this paper, we proposed a new variational model to learn an adaptive dictionary for sparse image modelling. Different from the K-SVD method, the dictionary learned in our approach is an orthogonal dictionary. The seemingly performance loss on sparse coding when adopting an orthogonal dictionary over an over-complete dictionary indeed has little negative impact on the performance of image restoration. The performance of the proposed orthogonal dictionary learning method is at least comparable to the K-SVD method in several image restoration applications. The gain by using an orthogonal dictionary is very noticeable. There exists a fast alternating iteration scheme for

solving the resulting variational model with rigorous justification on its optimality and convergence. In short, the proposed sparsity-based orthogonal dictionary learning method is much faster than the K-SVD method but with comparable performance in image restoration,

1.1. Motivation and our contributions

The main computational issue of the K-SVD method comes from the fact that the dictionary $\boldsymbol{D} \in \mathbb{R}^{n\times k}$ is a highly redundant dictionary ($k = 4n$ in [12]) which lacks additional constraints on the correlations among atoms. The purpose using such a high redundant dictionary is for maximizing the sparsity of the code $\{\boldsymbol{\alpha}_j\}$ by having more atoms in the dictionary. However, although a highly redundant dictionary allows the existence of more sparse codes, accurately estimating these code becomes less computationally feasible with the increased redundancy. One well-known measure on the quality of dictionary for sparse coding is the so-called *mutual incoherence* $\mu(\boldsymbol{D})$ ([11]) defined as

$$\mu(\boldsymbol{D}) = \max_{i\neq j} \frac{|\boldsymbol{d}_i^\top \boldsymbol{d}_j|}{\|\boldsymbol{d}_i\|_2\|\boldsymbol{d}_j\|_2}$$

which measures the correlations among the atoms. It is known in compressed sensing literatures that the mutual incoherence constant $\mu(\boldsymbol{D})$ need to be small enough to guarantee the performance of sparse coding when using matching pursuit methods (see e.g. [28]). However, the constant μ of the redundant dictionary obtained via the K-SVD method and its variations usually is not small, as no constraints on its mutual incoherence are imposed during dictionary updating. In other words, the sparse coding using a redundant dictionary with large $\mu(D)$ becomes not only computationally demanding, but also may not be optimal. The negative impact of the dictionary with large $\mu(D)$ has been noticed in various sparse coding based recognition systems; see e.g. [30, 14]. One solution is to simultaneously minimize the term $\mu(D)$ when update the dictionary which leads to a complex non-convex minimization problem.

Moreover, the ideal atoms of the learned dictionary should be those represents repetitive local image patterns. For natural images, the number of such repetitive local image patterns is not necessarily very large. If we use a very redundant dictionary, some atoms might be either highly similar to others, or play very little role in the presentation. For example, it is shown in [22] that the dimension of the dictionary learned by the K-SVD method for face images can be reduced by half without causing much performance loss. In summary, we argue that when learning a dictionary for sparsity-based image restoration, a highly redundant dictionary often is not necessary for having a good sparse approximation. Instead, a dictionary with little redundancy and with very small $\mu(D)$ could perform as efficient as the redundant ones. For example, when using patch size of 8×8, the dictionary size of K-SVD is four times of that of the proposed orthogonal dictionary. However, both the K-SVDs and ours use about 4.3K coefficients to achieve an approximation with PSNR=25dB to the image "Lena".

Based on the above discussions, we propose to use an orthogonal dictionary for sparsity-based dictionary learning in image restoration, which leads to the following minimization model:

$$\begin{aligned} \min_{\boldsymbol{D}\in\mathbb{R}^{n\times r},\boldsymbol{\alpha}_i\in\mathbb{R}^n} & \textstyle\sum_i \|\boldsymbol{g}_i - [\boldsymbol{A};\boldsymbol{D}]\boldsymbol{\alpha}_i\|_2^2 + \lambda_i\|\boldsymbol{\alpha}_i\|_0, \\ s.t. \quad & \boldsymbol{D}^\top\boldsymbol{D} = I_r; \boldsymbol{A}^\top\boldsymbol{D} = \boldsymbol{0}, \end{aligned} \tag{2}$$

where $\{\boldsymbol{g}_i\} \subset \mathbb{R}^n$ denotes the set of image patches collected from the input image, $\boldsymbol{\alpha}_i$ denotes the code of the patch $\boldsymbol{g}_i$. $\boldsymbol{D} = \{\boldsymbol{D}_i\}_{j=1}^r$ denotes the set of r atoms of the dictionary for learning, $\boldsymbol{A} \in \mathbb{R}^{n\times n-r}$ denotes the set of $n - r$ atoms either from experiences or from other sources (A is allowed to be empty). The adoption of an orthogonal dictionary will greatly simplify the computation of both dictionary updating and sparse coding. Indeed, we will show in the main body that both sparse coding and dictionary updating in our model have explicit solutions.

1.2. Related Work

This section roughly categorizes the sparsity-based dictionary design.

Analytic transform (*e.g.* [25, 7, 21, 8]). The image restorations under some transform with explicit analytic definition typically works on small image patches. The small image patches are projected onto the space spanned by the atoms of the given transform to yield a set of sparse coefficients. The widely used transforms include both the orthogonal ones (*e.g.* DCT [25], wavelet bases [7]) and the redundant ones such as tight frame [8] and its data-driven extension [4]). When using these transforms, the small coefficients are erased as they are dominated by noise. In practice, the patches are chosen with overlaps such as the image is processed in a sliding windows fashion, which can attenuate the possible beam artifacts along patch boundaries.

Learned dictionary (*e.g.* [12, 15, 20, 26, 24, 27]). In recent years, the concept of the adaptivity has been exploited to design the dictionary specifically optimized for the target image, the so-called dictionary learning. The earlier works [23, 15] learn the dictionary from the statistics of image features or patches to obtain a sparser representation of natural images. The pioneering K-SVD method [12] learns an over-complete dictionary as well as the sparse representations of the patches under that dictionary in an alternating minimization framework. Starting from the set of overlapping image patches collected from the input image, the K-SVD method alternatively iterates between two subproblems: sparse coding and dictionary updating. Both subproblems in [12] are based on heuristic greedy methods: the

sparse coding under the overcomplete dictionary is solved via orthogonal matching pursuit (OMP) and the dictionary is estimated via column-wise sequentially SVD updates.

Since the appearance of the K-SVD method, many approaches have been proposed to further improve it on computational efficiency and effectiveness in image restoration. An efficient implementation of the K-SVD method is developed in [26] which uses the Batch-OMP to accelerate the sparse coding and use two simple matrix-vector product to replace the SVD operation. The multi-scale generations of the K-SVD method are proposed in [20, 24] to further improve the performance by learning multiple dictionaries from different sets of image patches corresponding to different scales. In [27], the MC penalty is proposed to replace ℓ_0 norm for better performance and better numerical stability.

The K-SVD method and its variations are not only used for image restoration, they are also used in various recognition tasks. For example, a discriminative approach is proposed in [18] for image classification. Zhang and Li [30] generalized the K-SVD algorithm for face recognition. Jiang *et al*. [14] proposed a label consistent K-SVD for both face recognition and object recognition. Fore recognition, the term $\mu(D)$ is usually included in the minimization model when updating a dictionary to lower its mutual incoherence. Such an approach leads to a non-convex minimization problem.

Combination of dictionary learning and non-local approach (*e.g*. [10, 17]). The non-local approach such as BM3D [6] is another representative patch-based image restoration approach which groups the similar patches into a 3D array and filters the 3D array. Several methods have been proposed to combine the non-local approach and dictonary-learning for better performance in image restoration. For example, based on the groups of similar patches, the K-SVD method is used in [17], the local PCA-based method is used in [9, 29] and the PCA-based dictionary learning is used [10] for image denoising.

2. Fast orthogonal dictionary learning

Throughout the paper, the following mathematical notations are adopted for discussion. We use upper case bold letters for matrices, *e.g*. $\boldsymbol{X}$ and use lower case bold letter for the vectors, *e.g*. $\boldsymbol{X}_i$. The Frobenius norm of a matrix $\boldsymbol{X}$ is define as: $\|\boldsymbol{X}\|_F := (\sum_{i,j} |\boldsymbol{X}_{i,j}|^2)^{1/2}$. $\|\boldsymbol{X}\|_0$ denotes the number of nonzero entries in $\boldsymbol{X}$. The trace of a matrix $\boldsymbol{X}$ is defined as: $\mathrm{Tr}(\boldsymbol{X}) := \sum_k \boldsymbol{X}_{k,k}$. Let $\boldsymbol{X} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{Q}^\top$ be the singular value decomposition (SVD) for X. Given a vector $\boldsymbol{v} \in \mathbb{R}^n$, the *hard thresholding* operator $T_\lambda(\boldsymbol{v})$ is defined as $[T_\lambda \boldsymbol{v}]_i = \boldsymbol{v}_i$, if $|\boldsymbol{v}_i| > \lambda$ and 0 otherwise.

2.1. Problem formulation

Given an image g, let $\boldsymbol{G} = \{\boldsymbol{g}_1, \cdots, \boldsymbol{g}_m\} \in \mathbb{R}^{n\times m}$ denote the training set of image patches of size $\sqrt{n} \times \sqrt{n}$ collected from the image after vecterization. The image patches for the training can be selected randomly or regularly. Now we consider the sparse approximation problem for the set $\boldsymbol{G}$ under an orthogonal dictionary $\hat{\boldsymbol{D}} := [\boldsymbol{A}, \boldsymbol{D}] \in \mathbb{R}^{n\times n}$ whose columns refers to dictionary atoms. The dictionary has two sub-dictionaries in our implementation: one is $\boldsymbol{A} \in \mathbb{R}^{n\times n-r}$ which contains the input orthogonal atoms known as good ones from other sources; the other is $\boldsymbol{D} \in \mathbb{R}^{n\times r}$ which denotes the set of atoms need to be learned from the input image. The orthogonal constraint on the dictionary says that

$$\hat{\boldsymbol{D}}^\top \hat{\boldsymbol{D}} = I_n \Rightarrow \boldsymbol{A}^\top \boldsymbol{A} = I_{n-r}; \boldsymbol{D}^\top \boldsymbol{D} = I_r;\ \boldsymbol{A}^\top \boldsymbol{D} = \boldsymbol{0}.$$

We propose to learn the orthogonal dictionary $\boldsymbol{D}$ via solving the following minimization model

$$\begin{aligned} \min_{\boldsymbol{D}\in\mathbb{R}^{n\times r}, \boldsymbol{V}\in\mathbb{R}^{n\times m}} & \|\boldsymbol{G} - [\boldsymbol{A}, \boldsymbol{D}]\boldsymbol{V}\|_F^2 + \lambda^2 \|\boldsymbol{V}\|_0 \\ s.t. \quad & \boldsymbol{D}^\top \boldsymbol{D} = I_r; \boldsymbol{A}^\top \boldsymbol{D} = \boldsymbol{0}, \end{aligned} \tag{3}$$

It is noted that $r = n$ if the set $\boldsymbol{A}$ is empty.

The minimization (3) is quite similar to the model (1) used in the K-SVD method , except some additional linear and bi-linear constraints on $\boldsymbol{D}$. In the next, we will show that the minimization (3) is much easier to solve than (1).

2.2. Numerical method

Same as the K-SVD method, we take an alternating iterative scheme to solve (3). More specifically, let $\boldsymbol{D}^{(0)}$ be the initial dictionary to start (*e.g*. the DCT dictionary or multi-scale wavelet dictionary). Then for $k = 0, 1, \ldots, K-1$,

1. sparse coding: given the orthogonal dictionary $\boldsymbol{D}^{(k)}$, find the sparse code $\boldsymbol{V}^{(k)}$ via solving

$$\boldsymbol{V}^{(k)} := \arg\min_{\boldsymbol{V}\in\mathbb{R}^{n\times m}} \|\boldsymbol{G} - [\boldsymbol{A}, \boldsymbol{D}^{(k)}]\boldsymbol{V}\|_F^2 + \lambda^2 \|\boldsymbol{V}\|_0 \tag{4}$$

2. dictionary updating: given the sparse code $\boldsymbol{V}^{(k)}$, update the dictionary via solving the minimization:

$$\begin{aligned} \boldsymbol{D}^{(k+1)} := & \arg\min_{\boldsymbol{D}\in\mathbb{R}^{n\times r}} \|\boldsymbol{G} - [\boldsymbol{A}, \boldsymbol{D}]\boldsymbol{V}^{(k)}\|_F^2, \\ & \text{s.t.} \quad \boldsymbol{D}^\top \boldsymbol{D} = \boldsymbol{I}_r, \boldsymbol{A}^\top \boldsymbol{D} = \boldsymbol{0}. \end{aligned} \tag{5}$$

In the next, we show that both the minimization (4) for sparse coding and (5) for dictionary update are trivial to solve. Indeed, each of them has an explicit solution. Define $\hat{\boldsymbol{D}} = [\boldsymbol{A}, \boldsymbol{D}^{(k)}]$. Then by the definition of $\boldsymbol{A}$ and $\boldsymbol{D}^{(k)}$, we have $\hat{\boldsymbol{D}}^\top \hat{\boldsymbol{D}} = I_n$. The next proposition gives an explicit solution to (4).

Proposition 2.1 ***(sparse coding)*** *Suppose that $\hat{\boldsymbol{D}}^\top \hat{\boldsymbol{D}} = I_n$. The following minimization problem*

$$\min_{\boldsymbol{V}} \|\boldsymbol{G} - \hat{\boldsymbol{D}}\boldsymbol{V}\|_F^2 + \lambda^2 \|\boldsymbol{V}\|_0 \tag{6}$$

has a unique solution given by $\boldsymbol{V}^* = T_\lambda(\hat{\boldsymbol{D}}^\top \boldsymbol{G})$.

Proof See Appendix A.

For dictionary update, let $\boldsymbol{V}^{(k)} = [{\boldsymbol{V}_A^{(k)}}^\top, {\boldsymbol{V}_D^{(k)}}^\top]^\top$, where $\boldsymbol{V}_A^{(k)}$ denotes the codes associated with $\boldsymbol{A}$ and $\boldsymbol{V}_D^{(k)}$ associated with $\boldsymbol{D}^{(k)}$. Let $\mathcal{P}_A$ denote the orthogonal projection operator from $\mathbb{R}^n$ to the space spanned by the columns of $\boldsymbol{A}$: $\mathcal{P}_A \boldsymbol{v} = \boldsymbol{A}(\boldsymbol{A}^\top \boldsymbol{v}), \forall \boldsymbol{v} \in \mathbb{R}^n$. Then, the next proposition gives the explicit solution to the minimization (5).

Proposition 2.2 (dictionary updating) *The following constrained minimization*

$$\begin{aligned} &\min_{\boldsymbol{D}\in\mathbb{R}^{n\times r}} \|\boldsymbol{G} - (\boldsymbol{A}\boldsymbol{V}_A + \boldsymbol{D}\boldsymbol{V}_D)\|_F^2 \\ &\text{s.t. } \boldsymbol{D}^\top \boldsymbol{D} = \boldsymbol{I}_r,\ \boldsymbol{A}^\top \boldsymbol{D} = 0 \end{aligned} \tag{7}$$

has a unique solution given by $\boldsymbol{D}^* = \boldsymbol{P}\boldsymbol{Q}^\top$, where $\boldsymbol{P}$ and $\boldsymbol{Q}$ denote the orthogonal matrices defined by the following SVD

$$(\boldsymbol{I}_n - \mathcal{P}_A)\boldsymbol{G}\boldsymbol{V}_{\boldsymbol{D}}^\top = \boldsymbol{P}\Sigma\boldsymbol{Q}^\top.$$

Proof See Appendix B.

Therefore, each iteration in the proposed alternative iteration scheme is very simple. There is no need for solving any minimization problem when doing the sparse coding and dictionary updating. The sparse coding is done via a hard thresholding operation and the dictionary updating is done via a single SVD. See Algorithm 1 for the complete description of the algorithm.

Algorithm 1 Online orthogonal dictionary learning

Input: image patches $\boldsymbol{G}$, input orthogonal atoms $\boldsymbol{A}$
Output: learned dictionary $\boldsymbol{D}$
Main procedure:
1. Set the initial guess $\boldsymbol{D}^{(0)}$.
2. For $k = 0, 1, \ldots, K$,
   (a) $\boldsymbol{V}_D^{(k)} := T_\lambda({\boldsymbol{D}^{(k)}}^\top \boldsymbol{G})$;
   (b) run the SVD for the matrix

$$(\boldsymbol{I}_n - \mathcal{P}_A)\boldsymbol{G}{\boldsymbol{V}_D^{(k)}}^\top = \boldsymbol{P}\Sigma\boldsymbol{Q}^\top;$$

   (c) $\boldsymbol{D}^{(k+1)} := \boldsymbol{P}\boldsymbol{Q}^\top$.
3. $\boldsymbol{D} := \boldsymbol{D}^{(K+1)}$.

2.3. Complexity analysis of Alg. 1

In this section, we give a detailed analysis on the computational complexity of Algorithm 1 for sparsity-based orthogonal dictionary learning. Let m denotes the number of training patches in $\boldsymbol{G}$ and consider the worst scenario in which no pre-defined atom provided, i.e. $\boldsymbol{D} \in \mathbb{R}^{n\times n}$.

The sparse coding of Alg. 1 uses $2mn^2$ operations to obtain the matrix product $\boldsymbol{D}^\top\boldsymbol{G}$ and mn^2 operations in hard thresholding. Let K denote the average number of nonzero entries in each column of $\boldsymbol{V}$. For dictionary update of Alg. 1, the number of operations required to calculate the multiplications $\hat{\boldsymbol{G}}\boldsymbol{V}^\top$ is $2mnK$. The standard algorithm to obtain the singular value decomposition of $\hat{\boldsymbol{G}}\boldsymbol{V}^\top \in \mathbb{R}^{n\times n}$ takes $21n^3$ operations [13]. So, the total number of operations in one iteration of Alg. 1 is

$$\mathrm{T} = 3mn^2 + 2mnK + 21n^3 \tag{8}$$

	Approx. K-SVD [26]	Alg. 1
Sparse Coding	$m(8n^2 + 4K^2n + 12Kn) + mK^3 + 16n^3$	$3mn^2$
Dictionary Learning	$20mKn + 64n^3$	$2mKn + 21n^3$
Total	$m(8n^2 + 4nK^2 + 32Kn + K^3) + 80n^3$	$m(3n^2 + 2Kn) + 21n^3$

Table 1. Complexity analysis for one iteration

The K-SVD method [12] is very computationally demanding. The OMP used for sparse coding is known to be slow. The dictionary update of the K-SVD method need to call SVD operators for $4n$ times. Thus, a fast approximate K-SVD method is developed in [26] which use batch-OMP for sparse coding and replacing SVD by matrix-vector multiplication. The analysis of the approximate K-SVD method (the dimension of dictionary is set $4n$ by default), together with ours are listed in table 1. Clearly, Algorithm 1 requires far less operations. The computational efficiency in applications will be further investigated in the experiments.

2.4. Applications in image restoration

The sparsity-based online orthogonal dictionary learning Algorithm 1 is very simple to implement and also very computationally efficient. To evaluate its performance in image restoration in terms of recovery quality and computational efficiency, we applied Algorithm 1 on two sample image restoration tasks: image denoising and image in painting.

Image denoising. Algorithm 1 can be directly applied on de-noising by taking the noisy image as the input image for training. It is known in signal processing that most noise are in the high-pass channels. Thus, we fix a low-pass filter in the dictionary and only learn $n-1$ high-pass filters from the input image. That is, we define $\boldsymbol{A} = [\boldsymbol{\alpha}_0] \in \mathbb{R}^{n\times 1}$, where

$$\boldsymbol{\alpha}_0 = n^{-1/2}[1, 1, \ldots, 1]^\top.$$

Clearly, the orthogonal constraint $\boldsymbol{\alpha}_0^\top \boldsymbol{D} = 0$ on $\boldsymbol{D}$ ensures that all atoms in $\boldsymbol{D} \in \mathbb{R}^{n\times n-1}$ are high-pass filters. After generating the training matrix $\boldsymbol{G}$ by randomly sampling the image patches of size $\sqrt{n} \times \sqrt{n}$ from the noisy image, the dictionary D is learned from Algorithm 1. Then the de-noised image is reconstructed from the de-noised patch matrix $\hat{\boldsymbol{D}}T_{\lambda_1}(\hat{\boldsymbol{D}}^\top\boldsymbol{G})$ by averaging the overlaping pixels, where $\hat{\boldsymbol{D}} = [\boldsymbol{\alpha}_0, \boldsymbol{D}]$. For computationally efficiency, the

patches for trained are uniformly elected from the image at random. The patches for denoising are the patches uniformly selected with overlaps. See Algorithm 2 for details.

Image inpainting. Image in-painting is about recovering the missing values of image pixels or removing unwanted content from the image, which can be formulated as solving the following under-determined linear inverse problem:

$$f(k) = g(k) + \epsilon, \quad k \in \Lambda^c,$$

where g denote the image for recovery, Λ denotes the region for in-painting and Λ^c denotes its complement, and ϵ denotes noise. Using a dictionary $\boldsymbol{D}$ generated from wavelet frame filters, Cai *et al*. [3] proposed the following iteration scheme for in painting f:

$$\boldsymbol{G}^{(k+1)} = (I - P_\Lambda)\boldsymbol{F} + P_\Lambda \boldsymbol{D}^{-1}(T_\lambda \boldsymbol{D}\boldsymbol{G}^{(k)}), \quad (9)$$

where P_Λ is the diagonal projection matrix whose diagonal element equals to 1 if in Λ and 0 otherwise, $\boldsymbol{G}^{(k)}$ are mage patch matrices from $g^{(k)}$ and f respectively. In our implementation, we use the same iteration scheme. Different from image noising. During each iteration of Algorithm 1, we use the newest estimate $g^{(k)}$ to generate the training patch matrix. See Algorithm 3 for the details.

Algorithm 2 Denoising via orthogonal dictionary learning

Input: noisy image g
Output: denoised image g^*
Main procedure:
1. Initilization.
   (a) synthesizing image patch matrix $\boldsymbol{G}$ from g;
   (b) defining $\boldsymbol{A} = [\boldsymbol{a}_0]$ for some low-pass filter $\boldsymbol{a}_0$.
2. Learning a dictionary $\boldsymbol{D}^*$ using Algorithm 1 with input $\boldsymbol{G}$ and $\boldsymbol{A} = [\boldsymbol{a}_0]$.
3. De-noising patch matrix $\boldsymbol{G}^* := \hat{\boldsymbol{D}} T_{\lambda_1}(\hat{\boldsymbol{D}}^\top \hat{\boldsymbol{G}})$ with $\hat{\boldsymbol{D}} = [\boldsymbol{A}, \boldsymbol{D}^*]$.
4. Synthesizing the denoised image g^* from $\boldsymbol{G}^*$ by averaging the overlapping pixels.

3. Experiments

In this section, we evaluate the performance of the proposed orthogonal diction learning on image denoising and image in-painting. The experiments are conducted in MATLAB R2011b (64bit) Linux version on a PC workstation with an INTEL CPU (2.4GHZ) and 48G memory. The initial dictionary is generated by the local DCT transform: either 8×8 or 16×16. The image patches for training are uniformly selected from the input image at random. For image size 512×512 and patch size 16×16, about $4 * 10^4$ patches are used for training.

Computational efficiency. Under the same software and hardware environment, Algorithm 1 is compared to the widely used over-complete dictionary learning: the K-SVD algorithm [12] and its fast version, the approximated K-SVD algorithm [26] with the implementations from the original authors [1]. Table 3 listed the detailed running time of each module in K-SVD method, approximated K-SVD method and Agorithm 1. For each iteration, clearly Algorithm 1 is much faster than both the K-SVD method and the approximate K-SVD method.

The shorter running time for each iteration does not imply the algorithm run faster, as it might has slow convergence. Thus, we conduct another test on the overall running time when applying the three methods on image de-noising. The tested image is the image "Barbara" of 512×512 in the presence of i.i.d. Gaussian noise with s.t.d. $\sigma = 30$. Totally 15 iterations are used in the K-SVD method and the approximate K-SVD method as more iterations do no improve the PSNR value anymore. Table 3 listed the total running time of the two K-SVD methods and Alg. 1. While all three methods have comparable PSNR values, our method is much faster that the other two.

Image denoising. Algorithm 2 for image denoising is evaluated on several tested images shown in Fig. 2 with different

Algorithm 3 Inpainting via orthogonal dictionary learning

Input: image g and inpainting region Λ
Output: inpainted image g^*
Main procedure:
1. Initialization.
   (a) initilizing an in-painted image $g^{(0)}$ by interpolation;
   (b) synthesizing patch matrix $\boldsymbol{G}^{(0)}$ from $g^{(0)}$; and defining $\boldsymbol{A} = [\boldsymbol{a}_0]$.
2. For $k = 0, 1, \ldots, K$,
   (a) learning a dictionary $\boldsymbol{D}^{(k)}$ using one iteration of Algorithm 1 with input $\boldsymbol{G}^{(k)}$ and $\boldsymbol{A} = [\boldsymbol{a}_0]$;
   (b) synthesizing the image $h^{(k+1)}$ from the denoised patch matrix $\boldsymbol{G}^* := \hat{\boldsymbol{D}} T_{\lambda_1}(\hat{\boldsymbol{D}}^\top \boldsymbol{G})$;
   (c) defining $g^{(k+1)} := (I - P_\Lambda)(g) + P_\Lambda h^{(k+1)}$.
3. $g^* := g^{(K+1)}$.

implementation	module	8×8	16×16
K-SVD [12]	dictionary update	8.60	24.87
	sparse coding	1.19	2.18
Approx. K-SVD [26]	dictionary update	0.56	1.45
	sparse coding	1.44	3.50
Algorithm 1	dictionary update	0.02	0.15
	sparse coding	0.04	0.18

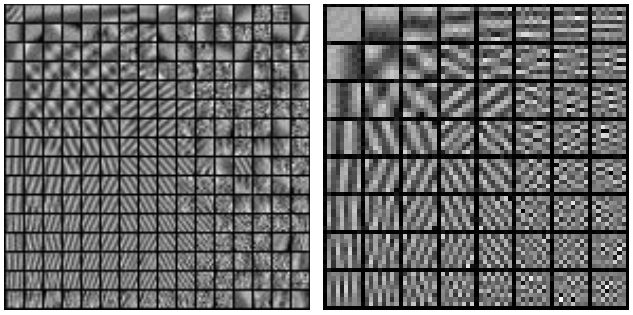
Table 2. Running time (in second) breakdown on one iteration of the K-SVD method, approximated K-SVD method and the implementation of Algorithm 1 with patch size 8×8 and 16×16.

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

Patch size\ method	running time (sec.) vs. PSNR (dB)	K-SVD	Approx. K-SVD	Alg. 1
8×8	time	202.75	98.35	2.02
	PSNR	28.51	28.61	28.44
16×16	time	484.25	206.49	12.11
	PSNR	27.86	27.84	28.93

Table 3. Running time of the K-SVD method, approximated K-SVD method with 15 iterations and Algorithm 1 with 30 iterations

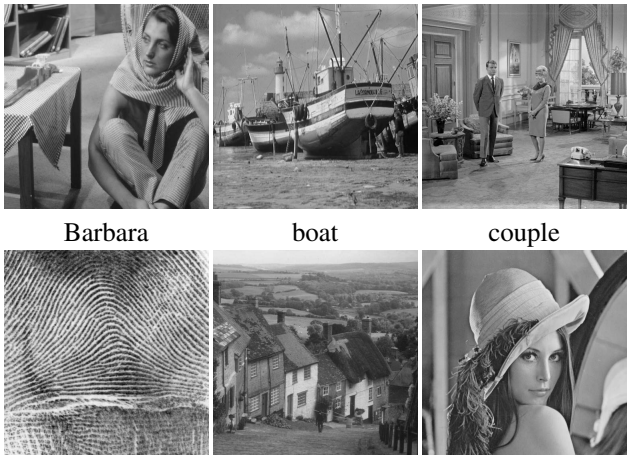
noise levels. Through all the experiments, we set $\lambda = 3.5\sigma$ and $\lambda_1 = 2.7\lambda$ as the thresholding value for the dictionary learning process. Our results are compared against two fixed transform based thresholding methods: linear spline framelet [8] and 8×8 DCT, the PCA-based non-local hierarchical method [9] and the K-SVD denoising method [12] with patch size of both 8×8 and 16×16. See Table 4 for the list of PSNR values of the results and Fig.1 for a visual illustration.



(e) K-SVD (f) Alg. 2

Figure 1. The dictionaries learned from the image "Barbara" with noise level $\sigma = 20$ using the K-SVD method and Algorithm 1. The size of atoms is 8×8. The number of dictionary atoms is 256 from the K-SVD method and is 64 from the proposed method.

Image Inpainting. Algorithm 3 is only tested on two sample image in-painting problems. The first example is the text removal from the image ([2]). The second example is to filling missing pixels in the image ([27]). In the first example, the results are compared to the classic in-painting method [2], and two dictionary learning based methods derived from the K-SVD method ([27]). The main difference between two dictionary learning methods lies in the choice of sparsity promoting functional: one uses the ℓ_1 norm and the other one uses MC penalty. The results are shown in Fig. 3, together with two zoom-in regions shown in the top-left and top-right corner of the image for easier inspection. It is seen that the result from Algorithm 3 has less artifacts than others. In the second example, the values of 50% of image pixels are missing at random. Algorithm 3 and two dictionary learning methods [27] are applied to recover the missing pixel values. See Figure. 4 for the visual illustration of the results. It is seen that Algorithm 3 outperformed the methods derived from the K-SVD methods.



Barbara boat couple

fingerprint hill lena

Figure 2. Test images.

4. Discussion and conclusion

In this paper, we proposed an orthogonal dictionary learning for image restoration, as an replacement of the widely used K-SVD method. The performance of the proposed orthogonal dictionary learning method is comparable to the K-SVD method, but it runs much faster than the K-SVD method. Such a significant improvement on the speed could be very important to many image restoration application when dealing with image of very large size or processing many images. In future, we would like to study how to effectively combine the non-local scheme and the proposed orthogonal dictionary learning method to develop better image restoration methods. Also, we will investigate the possible applications of the proposed methods in recognition.

Acknowledgement

The authors would like to thank the area chair and the reviewers for their helpful comments and suggestions. The work of J.-F. Cai was partially supported by a grant from the Simons Foundation (#281384 to J.-F. Cai) and the work of C. Bao and H. Ji was partially supported by Singapore MOE Research Grant R-146-000-165-112.

References

[1] M. Aharon and M. E. Bruckstein. K-svd: An algorithm for designing of overcomplete dictionaries for sparse representation. *IEEE Trans. Signal Processing*, 54(11), 2006.

[2] M. Bertalmio, G. Sapiro, V. Caselles, and C. Ballester. Image inpainting. In *ACM SIGGRAPH*, 2000.

[3] J. Cai, R. Chan, and Z. Shen. A framelet-based image inpainting algorith. *Appl. Comp. Harm. Anal.*, 24(2), 2008.

a b c d e

Figure 3. Comparison of text removal. (a) image with overlapped texts; (b-e) correspond to the results from [2], two over-complete dictionary learning method with ℓ_1 norm sparsity penalty and MC penalty ([27]), and Algorithm 3.

(a) b c d e

Figure 4. Image inpainting with 50% random missing pixels. (a) Original image; (b) corrupted image; (c-e) the results from from two over-complete dictionary learning method with ℓ_1 norm sparsity penalty and MC penalty ([27]), and Algorithm 3.

[4] J. Cai, S. Huang, H. Ji, Z. Shen, and G. Ye. Data-driven tight frame construction and image denoising. CAM report 12-40, UCLA, 2013.

[5] J. Cai, H. Ji, C. Liu, and Z. Shen. Framelet based blind image deblurring from a single image. *IEEE Trans. Image Processing*, 21(2):565–572, 2012.

[6] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian. Image denoising by sparse 3-d transform-domain collaborative filtering. *IEEE Trans. Image Processing*, 16(8):2080–2095, 2007.

[7] I. Daubechies. *Ten Lectures on Wavelets*. SIAM, 1992.

[8] I. Daubechies, B. Han, A. Ron, and Z. Shen. Framelets: Mra-based constructions of wavelet frames. *Appl. Comp. Harm. Anal.*, 14(1):1–46, 2003.

[9] C.-A. Deledalle, J. Salmon, and A. S. Dalalyan. Image denoising with patch based pca: local versus global. In *BMVC*, 2011.

[10] W. Dong, X. Li, L. Zhang, and G. Shi. Sparsity-based image denoising via dictionary learning and structured clustering. In *CVPR*, 2011.

[11] D. Donoho and M. Elad. Optimally sparse representation in general (non-orthogonal) dictionary via ℓ_1 minimization. *PNAS*, 100:2197–2202, 2002.

[12] M. Elad and M. Aharon. Image denoising via sparse and redundant representations over learned dictionaries. *IEEE Trans. Image Processing*, 15(12):3736–3745, 2006.

[13] G. Golub and C. V. Loan. *Matrix Computations*. JHU Press, 1996.

[14] Z. Jiang, Z. Lin, and L. Davis. Learning a dicscriminative dictionary for sparse coding via label consistent K-SVD. In *CVPR*, 2011.

[15] M. S. Lewicki and T. J. Sejnowski. Learning overcomplete representations. *Neural computation*, 12(2):337–365, 2000.

[16] J. Mairal, F. Bach, J. Ponce, and G. Sapiro. Online learning for matrix factorization and sparse coding. *JMLR*, 11, 2010.

[17] J. Mairal, F. Bach, J. Ponce, G. Sapiro, and A. Zisserman. Non-local sparse models for image restoration. In *ICCV*, pages 2272–2279. IEEE, 2009.

[18] J. Mairal, F. Bach, J. Ponce, G. Sapiro, and A. Zisserman. Supervised dictionary learning. In *NIPS*, 2009.

[19] J. Mairal, M. Elad, and G. Sapiro. Sparse representation for color image restoration. *Image Processing, IEEE Transactions on*, 17(1):53–69, 2008.

[20] J. Mairal, G. Sapiro, and M. Elad. Learning multiscale sparse representations for image and video restoration. Technical report, DTIC Document, 2007.

[21] S. Mallat. *A Wavelet Tour of Signal Processing: The Sparse Way*. Academic Press, third edition, 2008.

[22] R. Mazhar and P. D. Gader. EK-SVD: Optimized dictionary design for sparse representations. In *ICPR*, 2008.

[23] B. A. Olshausen, D. J. Field, et al. Sparse coding with an overcomplete basis set: A strategy employed by vi? *Vision research*, 37(23):3311–3326, 1997.

[24] B. Ophir, M. Lustig, and M. Elad. Multi-scale dictionary learning using wavelets. *IEEE J. Selected Topics in Signal Processing*, 5(5):1014–1024, 2011.

[25] K. Rao and P. Yip. *Discrete Cosine Transform: Algorithms, Advantages and Applications*. Academic Press, 1990.

[26] R. Rubinstein, M. Zibulevsky, and M. Elad. Efficient implementation of the k-svd algorithm using batch orthogonal matching pursuit. *CS Technion*, 2008.

Image	Babara					Boat					Couple				
σ	10	20	30	40	50	10	20	30	40	50	10	20	30	40	50
DCT; 8×8	34.13	30.24	27.96	26.41	25.15	33.49	30.01	27.96	26.51	25.42	33.41	29.86	27.79	26.33	25.25
wavelet frame	32.08	27.98	25.76	24.25	23.18	32.80	29.36	27.25	25.74	24.48	33.06	29.42	27.24	25.60	24.39
hierarchical PCA	34.52	30.85	28.92	27.38	26.00	33.65	30.23	28.24	26.75	25.57	33.56	29.95	27.86	26.41	25.32
K-SVD; 8×8	34.48	30.86	28.57	26.92	25.47	**33.67**	**30.41**	**28.44**	**27.04**	25.94	33.55	30.01	27.90	26.40	25.31
K-SVD; 16×16	34.09	30.27	27.81	26.09	24.78	33.06	29.48	27.28	25.89	24.86	32.87	29.10	26.85	25.19	24.08
Alg.2; 8	34.34	30.58	28.44	26.94	25.75	33.64	30.33	28.38	27.00	25.95	**33.57**	**30.04**	**28.06**	26.62	25.57
Alg.2; 16	**34.56**	**31.00**	**28.94**	**27.44**	**26.31**	33.51	30.26	28.36	27.00	**25.99**	33.40	29.97	28.05	**26.70**	**25.71**
Image	Fingerprint					Hill					Lena				
σ	10	20	30	40	50	10	20	30	40	50	10	20	30	40	50
DCT; 8×8	32.25	28.29	26.08	24.49	23.27	33.24	30.02	28.26	26.94	25.91	35.29	31.86	29.74	28.17	26.90
wavelet frame	30.44	26.49	24.26	22.70	21.45	32.69	29.46	27.58	26.12	24,96	34.22	30.69	28.52	26.83	25.50
hierarchical PCA	32.33	28.38	**26.31**	24.83	23.62	**33.41**	30.20	**28.59**	27.34	26.31	35.39	32.25	30.47	29.03	27.70
K-SVD; 8×8	**32.40**	**28.47**	26.29	24.70	23.19	33.38	30.20	28.39	27.15	26.28	**35.56**	**32.45**	30.49	29.03	27.82
K-SVD; 16×16	31.88	27.69	25.26	23.49	22.22	32.81	29.38	27.38	25.99	24.94	35.02	31.71	29.57	28.06	26.78
Alg.2; 8×8	32.24	28.33	26.17	24.68	23.47	33.27	**30.21**	28.51	27.31	26.43	35.52	32.31	30.32	28.84	27.66
Alg.2; 16×16	32.25	28.35	26.25	**24.86**	**23.81**	33.18	30.19	28.54	**27.40**	**26.54**	35.52	32.40	**30.49**	**29.09**	**27.95**

Table 4. PSNR values of the denoised results from different methods

[27] J. Shi, X. Ren, G. Dai, J. Wang, and Z. Zhang. A non-convex relaxation approach to sparse dictionary learning. In *CVPR*, 2011.

[28] A. Tropp. Greed is good: algorithmic results for sparse approximation. *IEEE Trans. Inf. Theory*, 50(10), 2004.

[29] L. Zhang, W. Dong, D. Zhang, and G. Shi. Two-stage image denoising by principal component analysis with local pixel grouping. *Pattern Recognition*, 43(4):1531–1549, 2010.

[30] Q. Zhang and B. Li. Discriminative k-SVD for dictionary learning in face recognition. In *CVPR*, 2010.

[31] H. Zou, T. Hastie, and R. Tibshirani. Sparse principal component analysis. *Journal of computational and graphical statistics*, 15(2):265–286, 2006.

Appendix A: Proof of Proposition 2.1.

By the fact that $\hat{\boldsymbol{D}}^\top \hat{\boldsymbol{D}} = I_n$, the minimization (6) is the equivalent to the following minimization

$$\min_{\boldsymbol{V}} \|\hat{\boldsymbol{D}}^\top \boldsymbol{G} - \boldsymbol{V}\|_F^2 + \lambda^2 \|\boldsymbol{V}\|_0, \tag{10}$$

which can rewritten as

$$\min_{\{\boldsymbol{V}_{i,j}\}} \sum_{ij} (\boldsymbol{V}_{i,j} - (\hat{\boldsymbol{D}}^\top \boldsymbol{G})_{i,j})^2 + \lambda |\boldsymbol{V}_{i,j}|$$

or equivalently the summation of multiple independent univariate minimization problems

$$\sum_{i,j} \min_{\{\boldsymbol{V}_{i,j}\}} (\boldsymbol{V}_{i,j} - (\hat{\boldsymbol{D}}^\top \boldsymbol{G})_{i,j})^2 + \lambda |\boldsymbol{V}_{i,j}|.$$

Recall that minimization problem $\min_{x \in \mathbb{R}} \|x - y\|_2^2 + \lambda^2 \|x\|_0$ has a unique solution $x^* = T_\lambda(y)$. Thus, the unique minimizer for (6) is $T_\lambda(\hat{\boldsymbol{D}}^\top \boldsymbol{G})$.

Appendix B: Proof of Proposition 2.2.

The objective function in (7) is equal to

$$\begin{aligned} &\|\boldsymbol{G} - \boldsymbol{A}\boldsymbol{V}_A - \boldsymbol{D}\boldsymbol{V}_D\|_F^2 \\ =&\|\boldsymbol{G} - \boldsymbol{A}\boldsymbol{V}_A\|_F^2 + \|\boldsymbol{D}\boldsymbol{V}_D\|_F^2 - \mathrm{Tr}((\boldsymbol{G} - \boldsymbol{A}\boldsymbol{V}_A)^\top \boldsymbol{D}\boldsymbol{V}_D). \end{aligned} \tag{11}$$

If $\boldsymbol{D}^\top \boldsymbol{D} = I$ and $\boldsymbol{A}^\top \boldsymbol{D} = \boldsymbol{0}$, then the first two terms in (11) are constant and $\mathrm{Tr}((\boldsymbol{A}\boldsymbol{V}_A)^\top \boldsymbol{D}\boldsymbol{V}_D) = 0$. Therefore, the minimization (7) is equivalent to

$$\max_{\boldsymbol{D}} \mathrm{Tr}(\boldsymbol{D}^\top \boldsymbol{G}\boldsymbol{V}_{\boldsymbol{D}}^\top), \text{s.t. } \boldsymbol{D}^\top \boldsymbol{D} = I_r, \boldsymbol{A}^\top \boldsymbol{D} = \boldsymbol{0}. \tag{12}$$

Considering the following SVD:

$$(\boldsymbol{I}_n - \mathcal{P}_A)\boldsymbol{G}\boldsymbol{V}_{\boldsymbol{D}}^\top = \boldsymbol{P}\Sigma\boldsymbol{Q}^\top.$$

From the Theorem 4 in [31], $\boldsymbol{D} = \boldsymbol{P}\boldsymbol{Q}^\top$ is the minimizer of the following minimization problem

$$\max_{\boldsymbol{D} \in \mathbb{R}^{n \times r}} \mathrm{Tr}(\boldsymbol{D}^\top (\boldsymbol{I} - \mathcal{P}_{\boldsymbol{A}})\boldsymbol{G}\boldsymbol{V}_{\boldsymbol{D}}^\top), \text{ s.t. } \boldsymbol{D}^\top \boldsymbol{D} = \boldsymbol{I}_r. \tag{13}$$

Notice that the space spanned by the columns $\boldsymbol{P}$ is equal to the one spanned by the columns of $(\boldsymbol{I} - \mathcal{P}_{\boldsymbol{A}})\boldsymbol{G}\boldsymbol{V}_{\boldsymbol{D}}^\top$ which is orthogonal to the space spanned by $\boldsymbol{A}$. Therefore, $\boldsymbol{A}^\top \boldsymbol{D} = \boldsymbol{A}^\top \boldsymbol{P}\boldsymbol{Q}^\top = \boldsymbol{0}$. Put all together, we have $\boldsymbol{D} = \boldsymbol{P}\boldsymbol{Q}^\top$ is the minimizer to the following minimization problem.

$$\begin{aligned} \max_{\boldsymbol{D} \in \mathbb{R}^{r \times p}} &\mathrm{Tr}(\boldsymbol{D}^\top (\boldsymbol{I} - \mathcal{P}_{\boldsymbol{A}})\boldsymbol{G}\boldsymbol{V}_{\boldsymbol{D}}^\top), \\ &\text{s.t. } \boldsymbol{D}^\top \boldsymbol{D} = \boldsymbol{I}_p, \boldsymbol{A}^\top \boldsymbol{D} = \boldsymbol{0}. \end{aligned} \tag{14}$$

Together with the fact

$$\begin{aligned} \boldsymbol{D}^\top \boldsymbol{G}\boldsymbol{V}_{\boldsymbol{D}}^\top &= \boldsymbol{D}^\top \mathcal{P}_{\boldsymbol{A}} \boldsymbol{G}\boldsymbol{V}_{\boldsymbol{D}}^\top + \boldsymbol{D}^\top (\boldsymbol{I} - \mathcal{P}_{\boldsymbol{A}})\boldsymbol{G}\boldsymbol{V}_{\boldsymbol{D}}^\top \\ &= \boldsymbol{D}^\top (\boldsymbol{I} - \mathcal{P}_{\boldsymbol{A}})\boldsymbol{G}\boldsymbol{V}_{\boldsymbol{D}}^\top \end{aligned} \tag{15}$$

The last equality in (15) holds when the constraint $\boldsymbol{A}^\top \boldsymbol{D} = \boldsymbol{0}$ is satisfied. The proof is complete.