Cascade Residuals Guided Nonlinear Dictionary Learning

Tong Zhang, Fatih Porikli

Research School of Engineering, Australian National University

Abstract

In this paper, we aim to extend dictionary learning onto hierarchical image representations in a principled way. To achieve dictionary atoms capture additional information from extended receptive fields and attain improved descriptive capacity, we present a two-pass multi-resolution cascade framework for dictionary learning and sparse coding. This cascade method allows collaborative reconstructions at different resolutions using only the same dimensional dictionary atoms. The jointly learned dictionary comprises atoms that adapt to the information available at the coarsest layer, where the support of atoms reaches a maximum range, and the residual images, where the supplementary details refine progressively a reconstruction objective. The residual at a layer is computed by the difference between the aggregated reconstructions of the previous layers and the downsampled original image at that layer. Our method generates flexible and accurate representations using only a small number of coefficients. It is computationally efficient since it encodes the image at the coarsest resolution while yielding very sparse residuals. Our extensive experiments on multiple image coding, denoising, inpainting and artifact removal tasks demonstrate that our method provides superior results.

Keywords: Sparse Coding, Dictionary Learning

Email addresses: `tong.zhang@anu.edu.au` (Tong Zhang), `fatih.porikli@anu.edu.au` (Fatih Porikli)

1. Introduction

Sparse representations of visual data promise several advantages including noise resilience by focusing on the consistently observed patterns in data distribution, improved classification performance by learning discriminative features, robustness by preventing the model from overfitting the training data, and semantic interpretation capability by allowing atoms to associate with meaningful attributes. As a result, they have been incorporated in many computer vision tasks such as compression, regularization in reverse problems, feature extraction, classification and recognition, interpolation for incomplete data, to count a few [1, 2, 3, 4, 5, 6].

An overcomplete dictionary that leads to a sparse representation of the input data can be constructed from a predetermined set of vectors (predetermined dictionary) in a way that is agnostic to the data. It can also be learned by adapting its atoms to the data samples (learned dictionary). The performance of the predetermined dictionaries, e.g., overcomplete bases of Discrete Cosine Transform (DCT) [7], wavelets [8], curvelets [9], contourlets [10], shearlets [11],etc., depends on how well these bases align with the distribution of data samples. In comparison, the learned dictionaries are derived from the given data, and they can be tailored to attain additional objectives. Noteworthy methods for obtaining learned dictionaries can be listed as the Method of Optimal Directions (MOD) [12], generalized PCA [13], KSVD [2], and Online Dictionary Learning (ODL) [14, 4]. By adapting the input data, the learned dictionaries provide improved performance.

In general, the dictionary learning and sparse encoding tasks for a given image can be formulated as a constrained optimization problem

$$\arg\min_{\mathbf{D},\mathbf{x}_i} \|\mathbf{y}_i - \mathbf{D}\mathbf{x}_i\|_F^2 \qquad \text{s.t. } \|\mathbf{x_i}\|_0 \leq T \ , \tag{1}$$

or its equivalent form,

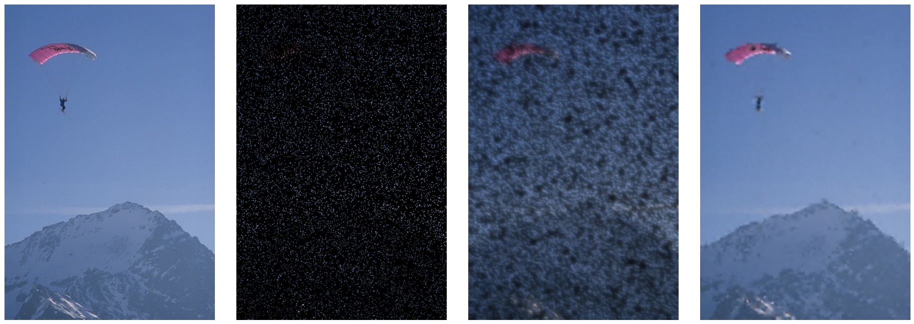
$$\arg\min_{\mathbf{D},\mathbf{x}} \sum_i \|\mathbf{x}_i\|_0 \qquad \text{s.t. } \|\mathbf{y_i} - \mathbf{D}\mathbf{x_i}\|_F^2 \leq \epsilon, \tag{2}$$

where the input data $\mathbf{y}_i \in \mathbb{R}^n$ are image patches of size $\sqrt{n}\times\sqrt{n}$, $\mathbf{x}_i \in \mathbb{R}^{m\times 1}$ denotes the corresponding representation of the i-th patch, $\mathbf{D} \in \mathbb{R}^{n\times m}$ is the overcomplete dictionary matrix where $m > n$, T is the number of the non-zero valued coefficients, and ϵ is the error tolerance on the reconstruction error. One fundamental aspect of this model is that the coefficient vector $\mathbf{x}_i$

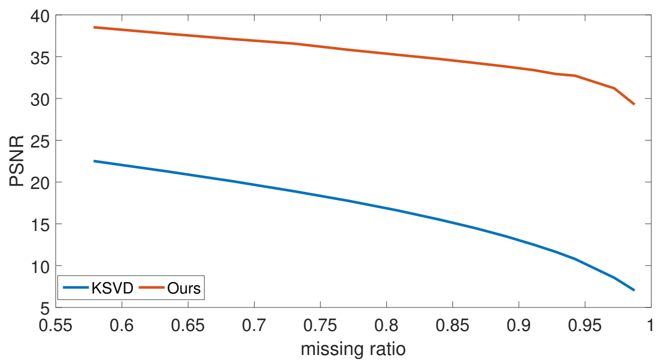
is sparse, in other words, $T \ll m$, which implies that the signal is composed of a few dictionary atoms. For an extended discussion on the solutions of the above objectives, please see Section 2.

Dictionary learning methods operate on dimensional vector spaces. For example, 8×8 image patches are represented by 64-dimensional vectors. The dimensionality of the vectors, thus the size of the patches, is required to be constant for the distance computations and the formulation of the optimization objectives. However, dictionary atoms obtained in this fashion are blind to larger context since they only see the local information contained within the constant size image patches. Simply increasing the patch size may extend the support area for contextual information yet it also decreases the flexibility of the dictionary to fit the data, puts a limit on the number of data samples and increases the computational complexity exponentially. Moreover, the optimal patch size may vary depending on the underlying information, e.g., visual texture, in the image. To attain the reconstruction error small while keeping the sparsity constraint low, a finer partitioning of the image by smaller patches would be preferable within the highly textured regions, yet larger blocks would result in improved sparsity for the smooth areas. Assume that we have a 256×256 image where all pixels have the same value. Using the conventional 8×8 overlapping patches we need more than 60K coefficients to encode the image, yet the same image can be represented using only a small number of coefficients of larger patches, even only a single coefficient in the ideal case of the patch is equal to the size of the image.

As a remedy, multi-scale dictionary learning methods aim to learn dictionaries at different image resolutions for the same patch size, e.g. using shearlets, wavelets, and Laplacian pyramid [4, 5, 15, 16, 17]. A drawback of these methods is that each layer in the pyramid is either processed independently or in small frequency (power spectrum) bands; thus the reconstruction errors of the coarser layers are projected directly onto the finest layer. Besides, this impedes compensation of such errors by and in the previous layers. Since the reconstruction error is correlated with the local texture, to attain a spatially consistent reconstruction, all layers need to be constructed accurately. Instead of learning in different image resolutions, [18] first builds a set of separate dictionaries for the quadtree partitioned patches and then it pads (with zeros) the smaller patches to the largest scale. However, the dimensionality of the dictionary learned in this fashion is still proportional to the maximum patch size, which brings increased computational load and memory requirements.



(a) Original (b) Corrupted (c) KSVD (d) Ours



(e) Reconstruction quality vs. ratio of missing pixels

Figure 1: (a) Original image. (b) Corrupt image where 93% of the original pixels are removed. (c) Reconstruction result of KSVD. PSNR is 11.80 dB. (d) Reconstruction result of our method. PSNR is 33.34 dB. (e) Reconstructed quality vs. the rate of missing pixels. As visible, our method is superior to KSVD.

Moreover, existing multi-scale dictionary learning methods often overlook the redundancy between the layers. As a consequence, in addition to requiring larger dimensional dictionaries, a high number of coefficients are spent unnecessarily on the smooth areas due to lack of communication between the layers. To the best of our knowledge, no conventional method offers a systematic solution where encodings of the coarser scales progressively enhance the reconstruction results of the finer layers.

Our Contributions

We present a computationally efficient framework that employs multi-resolution residual maps for dictionary learning and sparse coding in order to address the above shortcomings and allow dictionary atoms to access larger context for an improved descriptive capacity.

To this end, we start with building an image pyramid using bicubic interpolation. In the first pass, we learn a dictionary from the coarsest resolution layer and obtain the sparse representation. We upsample the reconstructed image and compute the residual in the next layer. The residual at a level is computed by the difference between the aggregated reconstructions from the coarser layers in a cascade fashion and the downsampled original image at that layer. Dictionaries are learned from the residual in every layer. We use the same patch size yet different resolution input images, which is instrumental in reducing computations and capturing larger context through. The computational efficiency stems from encoding at the coarsest resolution and encoding the residuals that are significantly sparse. This enables our cascade to go as deep as needed without any compromise.

In the second pass, we collect all patches from all cascade layers and learn a single dictionary for a final encoding. This naturally solves the problem of determining how many atoms to be assigned at a hierarchical layer. Thus, all atoms in the dictionary have the same dimensionality while their receptive fields vary depending on the layer.

Compared to existing multi-scale approaches operating indiscriminately on image pyramids or wavelets, our dictionary comprises atoms that adapt to the information available at each layer. The details learned from residual images progressively refine our reconstruction objective. This allows our method to generate a flexible image representation using much smaller number of coefficients. Our extensive experiments demonstrate that our method applies favorably in image coding, denoising, inpainting and artifact removal tasks. Figure 1 shows an inpainting result generated by our method where the input image was missing 93% of its pixels. As visible, we can recover even the very large areas of missing pixels.

2. Related Work

The nature of the dictionary learning objective makes it an NP-hard problem since neither the dictionary nor the coefficients are known. To handle this challenge, most dictionary learning algorithms alternate between the sparse coding and dictionary updating steps iteratively by fixing one while optimizing the other. For example, MOD updates the dictionary by solving an analytic solution of the quadratic problem by using Moore-Penrose pseudo-inverse; KSVD incorporates the k-means clustering and singular value decomposition by refining the coefficients and dictionary atoms recursively; and ODL updates the dictionary by using the first-order stochastic gradient descent in small batches. Adding to the complexity, sparse coding itself is an NP-hard problem due to the ℓ_0 norm. This objective is often approximated by greedy schemes such as Matching pursuit (MP) [19] and Orthogonal Matching Pursuit (OMP) [20]. Another alternative is to replace the ℓ^0-norm with the ℓ^p-norm with $p \leq 1$. When $p = 1$, the solution can be approximated by Basis Pursuit(BP) [21], FOCal Under-determined System Solver (FOCUSS) [22], and Least Angle Regression (LARS) [23] to count a few.

Multi-scale methods for image encoding have been widely studied in the past. Wavelets are among the premier multi-scale analysis tools in signal processing. Many wavelets variants, e.g., bandlets [24], contourlets [10], curvelets [9] as well as decomposition methods, e.g., wavelet pyramid [25], steerable pyramid [26], and Laplacian pyramid [27] have also been proposed. These methods basically improve the frequency-based analysis of Fourier transform by incorporating scale and spatial information.

There have been few attempts to learn multi-scale dictionaries. In [18], a quadtree structure is proposed. Dictionaries with different atom dimensions are obtained for different levels of the quadtree and then concatenated together by zero-padding smaller atoms in a dyadic fashion. Unfortunately, the number of scales and the maximum dimension of dictionary atoms are restricted due to the heavy computational and memory requirements. Besides, this approach ignores the coarse-scale information that may be more suitable to represent patches using atoms of the same size.

To overcome the computational issues, [5] extracts sub-dictionaries in the wavelet transform domain by exploiting the sparsity between the wavelets coefficients. This work leverages frequency selectivity of the individual levels of a wavelet pyramid to remove redundancy in the learned representations. Since separate dictionaries are learned for directional subbands, its performance is hampered in comparison to the single-scale KSVD for image denoising tasks. Their following work [6] exploits multi-scale analysis and single-scale dictionary learning, fusing both outputs by using a weighted joint sparse coding. Since the fused dictionary is several times larger than its single-scale version, the computational complexity is high. Besides, its denoising performance is sensitive to the size and category of images. A similar work [4] builds multi-resolution dictionaries on the wavelet pyramid by employing the k-means clustering before the ODL step. For each resolution, it clusters the patches of three subbands and then concatenates all dictionary atoms. Although its denoising performance improves due to non-local clustering on the image subbands, each layer requires a large dictionary, which reflects adversely on the computational load.

Multi-resolution sparse representations are also employed for image fusion and super-resolution. Given a pre-trained dictionary, [16] fuses two images by obtaining sparse coefficients for high-pass and low-pass frequency bands and applying OMP. The fused coefficient columns in each band are chosen by maximal ℓ_1 norm of corresponding coefficients. Towards the same goal, [17] merges two coefficient vectors; however, the fused coefficient columns are selected by ℓ_2 norm. Instead of training subdictionaries independently, it learns $3S+1$ subdictionaries jointly (S stands for the number of layers), which means the dimension of the matrix is $(3S+1)n \times k$ thus the learning stage is computationally expensive. In [15], authors propose a multi-scale approach to super-resolve the diffusion weighted images where the low-resolution dictionary is based on the shearlet transform and the high-resolution one is based on image intensity. In [28], a sparse representation is used to build a model for image interpolation. This model describes each patch as a linear combination of similar non-local patch neighbors, and every patch is represented with a specific dictionary. To decrease the coherence of the representation basis, it clusters patches into multiple groups and learns multiple local PCA dictionaries.

3. Sparse Coding on Cascade Layers

As mentioned above, previous dictionary learning algorithms often formulate the problem at hand using a linear model on a fixed dimension thus on a fixed patch scale, which hinders exploiting dictionary atoms in their full potential. In comparison, our approach is nonlinear due to its recursive nature where we encode the resulting residuals of the layers in previous hierarchical levels. In a single layer, we represent the current vector as a linear combination of dictionary atoms, where we keep the same as single layer sparse coding. After each layer, the representations are accumulated into the final reconstruction at the end. Let $\hat{\mathbf{Y}}'_n$ denote the estimated n-th layer and $\hat{\mathbf{Y}}$ denote the reconstructed image, then the overall process can be described as

$$\hat{\mathbf{Y}} = \hat{\mathbf{Y}}'_0 + \mathbf{U}(\hat{\mathbf{Y}}'_1 + \mathbf{U}(\hat{\mathbf{Y}}'_2 + ... + \mathbf{U}(\hat{\mathbf{Y}}'_N))), \tag{3}$$

where $\mathbf{U}$ is an upsampling function.

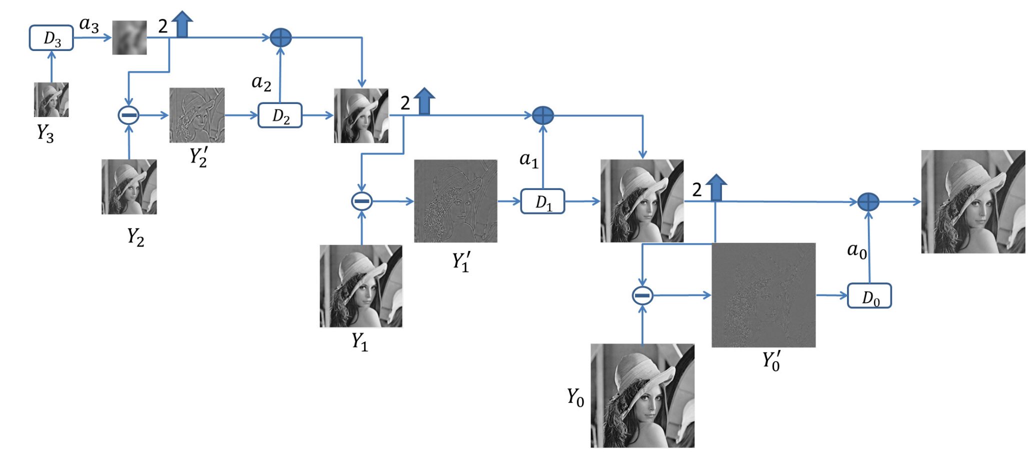


Figure 2: The first pass of our method for a 4-layer cascade. $\mathbf{Y}_0$ is the original image, $\{\mathbf{Y}_3, ..., \mathbf{Y}_0\}$ denote each layer of the image $\mathbf{Y}_3$ pyramid, and $\{\mathbf{D}_3, ..., \mathbf{D}_0\}$ are the dictionaries. $\mathbf{D}_3$ is learned from the downsampled image $\mathbf{Y}_3$ and the remaining dictionaries are learned from the residuals $\{\mathbf{Y}_2^{'}, \mathbf{Y}_1^{'}, \mathbf{Y}_0^{'}\}$. α_n are the reconstruction coefficients corresponding to the residual layers $\mathbf{Y}_n^{'}$.

A flow diagram of our framework is shown in Fig. 2 for a sample 4-layer cascade, where the input is a 512×512 grayscale image $\mathbf{Y}$. We first construct an image pyramid $\mathbf{Y} = \{\mathbf{Y}_0, \mathbf{Y}_1, ...\mathbf{Y}_N\}$ by bicubic downsampling. Here, $\mathbf{Y}_0$ is the finest (original) resolution and $\mathbf{Y}_N$ is the coarsest resolution. Other options for the image pyramid would be Gaussian pyramid, Laplacian pyramid, bilinear interpolation, and subsampling. Images resampled with bicubic interpolation are smoother and have fewer interpolation artifacts.

We employ a two-pass scheme where in the first pass we obtain residuals from layer-wise dictionaries, and in the second pass, we learn a single global dictionary that extracts and refines the atoms of the dictionaries generated in the first pass.

3.1. First Pass

We start at the coarsest (N-th) layer in the cascade. After learning the layer dictionary and finding the sparse coefficients, we propagate consecutively the reconstructed images to the finer layers. In the coarsest layer, we process the downsampled image. In the consecutive layers, we encode and decode the residuals. In each layer, we keep the size of image patches identical, which enable that a $b \times b$ patch in n-th layer corresponds to a $(2^n b) \times (2^n b)$ area in the original image. Algorithm 1 summarizes the first pass.

Algorithm 1 Cascade Sparse Coding

Input:

N (the highest pyramid layer), $\mathbf{Y}$(image),

T_n (number of coefficient used in layer n)

Output: $\mathbf{Y}', \hat{\mathbf{Y}}, \hat{\mathbf{D}}_{global}$

$\mathbf{Y}_n \leftarrow \text{subsampling}(\mathbf{Y}, 2^n)$

for $n = \{N, N-1, \cdots, 0\}$ **do**

  if n = N **then**

    $\mathbf{Y}_n' \leftarrow \mathbf{Y}_n$

  else

    $\mathbf{Y}_n' \leftarrow \mathbf{Y}_n - \text{upsample}(\hat{\mathbf{Y}}_{n+1}, 2)$

  Perform KSVD to learn dictionary $\hat{\mathbf{D}}_n$ and encode $\mathbf{Y}_n'$

  $\forall ij \ \{\hat{\mathbf{x}}_n^{ij}, \hat{\mathbf{D}}_n\} \leftarrow \underset{\mathbf{x}_n^{ij}, \mathbf{D}_n}{\arg\min} \sum_{ij} \|\mathbf{R}_{ij}\mathbf{Y}_n' - \mathbf{D}_n \mathbf{x}_n^{ij}\|_2^2 \quad \text{s.t } \|\mathbf{x}_n^{ij}\|_0 \leq T_n$

  if n = N **then**

    $\hat{\mathbf{Y}}_n \leftarrow (\sum_{ij} \mathbf{R}_{ij}^T \mathbf{R}_{ij})^{-1} (\sum_{ij} \mathbf{R}_{ij}^T \hat{\mathbf{D}}_n \hat{\mathbf{x}}_n^{ij})$

  else

    $\hat{\mathbf{Y}}_n \leftarrow (\sum_{ij} \mathbf{R}_{ij}^T \mathbf{R}_{ij})^{-1} (\sum_{ij} \mathbf{R}_{ij}^T \hat{\mathbf{D}}_n \hat{\mathbf{x}}_n^{ij}) + \text{upsample}(\hat{\mathbf{Y}}_{n+1}, 2)$

$\mathbf{Y}' \leftarrow \{\mathbf{Y}_N', \mathbf{Y}_{N-1}' \cdots, \mathbf{Y}_0'\}$

$\forall ij \ \hat{\mathbf{D}}_{\text{global}} \leftarrow \underset{\mathbf{D}}{\arg\min} \sum_{ij} \|\mathbf{R}_{ij}\mathbf{Y}' - \mathbf{D}\mathbf{x}^{ij}\|_2^2 \quad \text{s.t } \|\mathbf{x}^{ij}\|_0 \leq T$

Reconstruction:

$\hat{\mathbf{Y}} \leftarrow 0$

for $n = \{N, N-1, \cdots, 0\}$ **do**

  $\mathbf{Y}_n' = \mathbf{Y}_n - \text{upsample}(\hat{\mathbf{Y}}, 2)$

  $\forall ij \ \{\hat{\mathbf{x}}_n^{ij}\} \leftarrow \underset{\mathbf{x}_n^{ij}}{\arg\min} \sum_{ij} \|\mathbf{R}_{ij}\mathbf{Y}_n' - \hat{\mathbf{D}}_{global} \mathbf{x}_n^{ij}\|_2^2 \quad \text{s.t } \|\mathbf{x}_n^{ij}\|_0 \leq T_n$

  $\hat{\mathbf{Y}} \leftarrow (\sum_{ij} \mathbf{R}_{ij}^T \mathbf{R}_{ij})^{-1} (\sum_{ij} \mathbf{R}_{ij}^T \hat{\mathbf{D}}_{global} \hat{\mathbf{x}}_n^{ij}) + \text{upsample}(\hat{\mathbf{Y}}, 2)$

return

Dictionary Learning: We learn a dictionary at the coarsest layer and use it to reconstruct the downsampled image. This layer's dictionary $\hat{\mathbf{D}}_N$ is produced by minimizing the objective function using the coarsest resolution image patches

$$\arg\min_{\mathbf{D}_N, \mathbf{x}_N^{ij}} \sum_{ij} \|\mathbf{R}_{ij}\mathbf{Y}_N - \mathbf{D}_N \mathbf{x}_N^{ij}\|_2^2 + \lambda \|\mathbf{x}_N^{ij}\|_0 \tag{4}$$

where the operator $\mathbf{R}_{ij}$ is a binary matrix that extracts a square patch of size $b \times b$ at location (i, j) in the image then arranges the patch pixels into a column vector form. The parameter λ trades off the data fidelity term and the regularization term, and $\mathbf{x}_N^{ij}$ denotes the coefficients for the patch (i, j) .

In Fig. (9), we compare the efficiency of different learning algorithms. As shown, KSVD [2] underperforms in comparison to a-KSVD [29] and ODL [14] where both ODL and a-KSVD achieve the same PSNR with fewer coefficients. Our method does not assume a specific dictionary learning technique, and it can use any dictionary learning technique regardless of the way they update dictionary atoms. To demonstrate that our quality and sparsity improvements are not simply due to a specific choice of dictionary learning method, we employ the relatively handicapped and underperforming method, the original KSVD, to obtain our dictionaries. We initialize the dictionary $\mathbf{D}_N$ with a DCT basis by extracting several atoms from the DCT basis and then applying Kronecker product on the atoms to generate an overcomplete matrix, which is similar to KSVD. Notice that using a more efficient initialization scheme may produce better results and improve convergence [6].

During the dictionary learning stage, we fix all coefficient vectors $\mathbf{x}_N^{ij}$ and iteratively select dictionary atoms $\mathbf{d}_N^l$ one by one, $l = \{1, 2, \cdots, k\}$. For each atom $\mathbf{d}_N^l$, we extract the patches that are composed by the atom $(i, j) \in \mathbf{d}_N^l$ to compute the corresponding residual without the atom $\mathbf{d}_N^l$. The coefficients are denoted as $\mathbf{x}_N^{ij}(l)$, which are the non-zero entries of the l-th row of the coefficient matrix

$$\mathbf{e}_N^{ij}(l) = \mathbf{R}_{ij}\mathbf{Y}_N - \hat{\mathbf{D}}_N \mathbf{x}_N^{ij} + \mathbf{d}_N^l \mathbf{x}_N^{ij}(l). \tag{5}$$

Then, we arrange all $\mathbf{e}_N^{ij}(l)$ as the columns of the overall representation error matrix $\mathbf{E}_N^l$. We update the atom $\hat{\mathbf{d}}_N^l$ and the l-th row of coefficient matrix $\hat{\mathbf{x}}_N(l)$ by solving the equation

$$\{\hat{\mathbf{d}}_N^l, \hat{\mathbf{x}}_N(l)\} = \arg\min_{\mathbf{d},\mathbf{x}} \|\mathbf{E}_N^l - \mathbf{d}\mathbf{x}\|_F^2. \tag{6}$$

Finally, we perform a SVD decomposition on the error matrix, and update the l-th dictionary atom $\hat{\mathbf{d}}_N^l$ by the first column of $\mathbf{U}$, where $\mathbf{E}_N^l = \mathbf{U}\Sigma\mathbf{V}^T$; the coefficient vector $\hat{\mathbf{x}}_N(l)$ is replaced by the first column of matrix $\Sigma(1,1)\mathbf{V}$. In every iteration, all atoms and coefficients are updated simultaneously.

Sparse Coding: After obtaining the updated dictionary, sparse coding is employed with the Orthogonal Matching Pursuit (OMP), which is a computationally efficient greedy algorithm [30]. The sparse coding stops when the number of the non-zero coefficients reaches the upper limit T_N, or the reconstruction error becomes less than the threshold value, which depends on the specific task in hand. We update the coefficient vector $\hat{\mathbf{x}}_N^{ij}$ as

$$\hat{\mathbf{x}}_N^{ij} = \arg\min_{\mathbf{x}_N^{ij}} \sum_{ij} \|\mathbf{R}_{ij}\mathbf{Y}_N^{'} - \hat{\mathbf{D}}_N\mathbf{x}_N^{ij}\|_2^2 \quad \text{s.t.} \quad \|\mathbf{x}_N^{ij}\|_0 \leq T_N \tag{7}$$

and put it back into the dictionary learning stage to update the dictionary atoms and the coefficients.

Residuals: In each layer, we use at most T_n active coefficients for each patch to reconstruct the image and then compute the residual. The number of coefficients governs how strong the residual should emerge. Larger values of T_n favors for more accurate reconstructions; thus the total energy of residuals will decay. Smaller values of T_n cause the residual to increase, not only due to sparse coding but also resampling across layers. Since the dictionary is designed to represent a broad spectrum of patterns to keep the encodings as sparse as possible, T_n should be small. The reconstructed image is a weighted average of the patches that contain the same pixel

$$\hat{\mathbf{Y}}_N = (\sum_{ij} \mathbf{R}_{ij}^T\mathbf{R}_{ij})^{-1}(\sum_{ij} \mathbf{R}_{ij}^T\hat{\mathbf{D}}_N\hat{\mathbf{x}}_N^{ij}). \tag{8}$$

After decoding based on the dictionary $\hat{\mathbf{D}}_N$, we obtain the residual image $\mathbf{Y}_{N-1}^{'}$ by subtracting the upsampled reconstruction $\mathbf{U}(\hat{\mathbf{Y}}_N)$ from the next layer image $\mathbf{Y}_{N-1}$, e.g. $\mathbf{Y}_{N-1}^{'} = \mathbf{Y}_{N-1} - \mathbf{U}(\hat{\mathbf{Y}}_N)$. Here, $\mathbf{U}(\cdot)$ denotes the bicubic upsampling operator. Similar to the above dictionary learning and sparse coding procedure for the N-th layer, we reconstruct the residual $\hat{\mathbf{Y}}_{N-1}^{'}$ by training a separate residual dictionary $\mathbf{D}_{N-1}$ from the residual image itself. We keep encoding and decoding on residuals up to the finest layer. The procedure for the cascade residual dictionary learning and reconstruction can be expressed as follows

$$\{\hat{\mathbf{x}}_n^{ij}, \hat{\mathbf{D}}_n\} = \underset{\mathbf{x}_n^{ij}, \mathbf{D}_n}{\arg\min} \sum_{ij} \|\mathbf{R}_{ij}\mathbf{Y}_n^{'} - \mathbf{D}_n\mathbf{x}_n^{ij}\|_2^2 \quad \text{s.t.} \quad \|\mathbf{x}_n^{ij}\|_0 \leq T_n, \tag{9}$$

where residual image is

$$\mathbf{Y}_n^{'} = \begin{cases} \mathbf{Y}_n - \mathbf{U}(\hat{\mathbf{Y}}_{n+1}), & 0 \leq n < N \\ \mathbf{Y}_N, & n = N, \end{cases} \tag{10}$$

and the reconstructed residual is

$$\hat{\mathbf{Y}}_n = \begin{cases} (\sum_{ij}\mathbf{R}_{ij}^T\mathbf{R}_{ij})^{-1}(\sum_{ij}\mathbf{R}_{ij}^T\hat{\mathbf{D}}_n\hat{\mathbf{x}}_n^{ij}) + \mathbf{U}(\hat{\mathbf{Y}}_{n+1}), & 0 \leq n < N \\ (\sum_{ij}\mathbf{R}_{ij}^T\mathbf{R}_{ij})^{-1}(\sum_{ij}\mathbf{R}_{ij}^T\hat{\mathbf{D}}_n\hat{\mathbf{x}}_n^{ij}) & n = N. \end{cases} \tag{11}$$

Above, Eqn. 9 computes the coefficients with respect to the corresponding patches, and Eqn. 10 reconstructs the residual image for the next finer layer by subtracting the upsampled version of the coarser layer image from the image pyramid of the given layer. Similarly, Eqn. 11 is the general formulation of how we progressively reconstruct the image by adding the estimated residual and the upsampled image from the coarser layers.

Increasing the number of non-zero coefficients can reduce the error caused by the sparse representation. There is a trade-off between the number of coefficients and the quality of the reconstructed image. Our goal is to use the minimal number of coefficients while reconstructing an image of highest quality.

3.2. Second Pass

In each layer, the more atoms we use, the better quality can be achieved. However, this would not be the best use of the limited number of atoms. For instance, image patches from the coarsest layer are limited both in quantity and variety. The residual images are relatively sparse which imply they do not require many dictionary atoms. However, it is not straightforward to determine the optimal number of atoms for each dictionary since the finer level residuals depend heavily on the coarser ones.

Rather than keeping all dictionaries, we train a global dictionary $\mathbf{D}$ using patches from $\mathbf{Y}^{'} = \{\mathbf{Y}_N, \mathbf{Y}_{N-1}^{'}, \cdots, \mathbf{Y}_0^{'}\}$. As illustrated in Fig. 3.2, the dictionaries learned from $\mathbf{Y}^{'}$ in the first pass are redundant. The overall

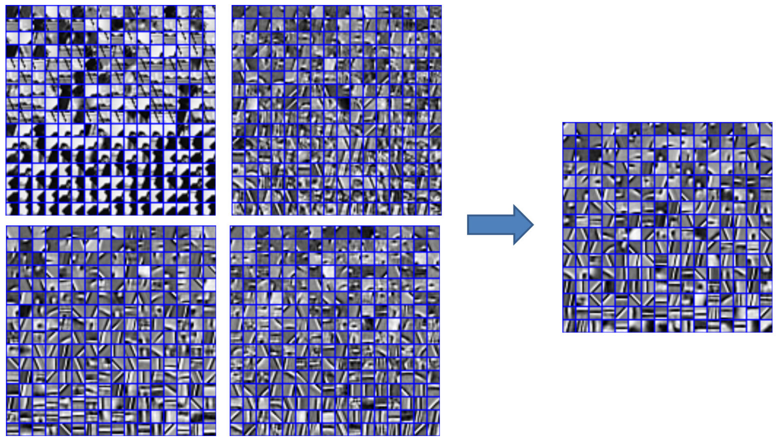


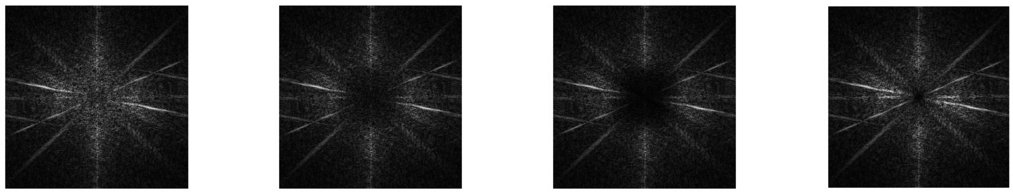
Figure 3: Left: The dictionaries learned in the first pass for the different levels (clockwise from the upper left: the coarsest level, the second level, the third level, and the finest level). Right: The unifying dictionary learned in the second pass.

dictionary is less repetitive thus more effective to reconstruct all four layers. Using a unified dictionary allows us to select most useful atoms automatically without making possibly suboptimal layer-wise decisions. Notice that, in this procedure, the number of coefficients can be chosen depending on the target quality of each layer.

4. Analysis

4.1. Role of the First Pass

The goal of the second pass is to find a unified and compact dictionary that is suitable for the reconstruction of all layers. From the coarsest to the finest layer, our algorithm reconstructs the input images at each layer. In the coarsest layer, the input image is a thumbnail version of the original image. In the following layers, the images correspond to the residuals between the reconstructed images and the scaled version of the original image. Our layers, except the coarsest one, are different from the corresponding Laplacian pyramid layers. To visualize this, we show the frequency domain versions of the residual in the finest layer for different levels of sparsity (1, 4, 10) applied to all other layers in Fig. 4. We also show the frequency transform of the finest level Laplacian pyramid image. As visible, using a higher number of coefficients in our method yields smaller residuals, in particular, the low-frequency components are more accurately reconstructed. When the sparsity



$T_n = 1$ $T_n = 4$ $T_n = 10$ Laplacian

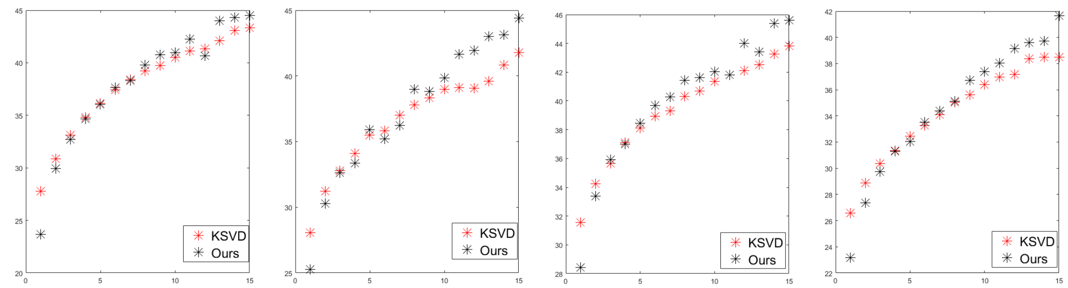
Figure 4: Residuals of the finest layer in the frequency domain for different values of coefficients used for each patch of Cameraman image is as the input. Right most is the Laplacian pyramid layer of the finest resolution. As visible, our method generates different layers depending on the sparsity level.

level is 1, the finest level image we obtain with our method 4-a and the Laplacian pyramid 4-d seem similar, yet as the sparsity level increases, their difference dilates significantly. If we learn a dictionary using the Laplacian pyramid and encode all layers using one coefficient per patch, the PSNR is 0.2 dB smaller than our hierarchical method. The PSNR will be less than 1 dB in case our method uses 10 coefficients per patch. These show that our residuals and Laplacian pyramid have different characteristics. Also, the residual pyramid generated by our method in the first pass plays a critical role in the reconstruction performance.

4.2. Second Pass: Generating a Unified Dictionary

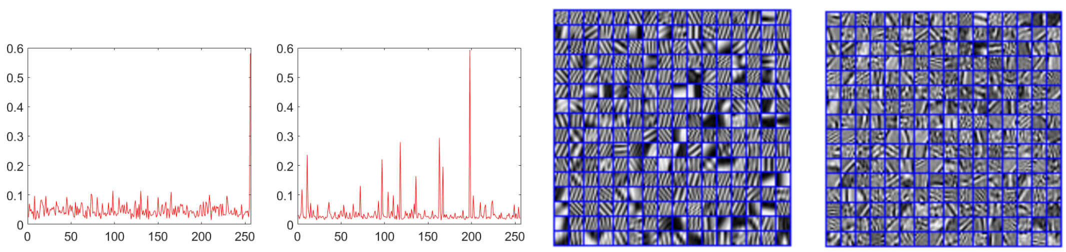
The nonconvex nature of the optimization algorithm for dictionary learning, i.e., updating the steps of learning the dictionary and then the corresponding sparse coefficients in a loop, may cause the solution to converge into one of the local minima. In our method, we utilize the OMP for sparse encoding, which is a greedy algorithm that does not guarantee the global minimum. Although we are seeking for a linear model for every layer, the final dictionary is based on the dictionaries of the previous layers. Thus, the solution we obtain can be regarded as a combination of the previous local minima.

To assess which dictionary learning method provides a higher reconstruction performance, we compare the reconstruction power of the dictionaries learned by the original KSVD method and our algorithm. We reconstruct the same single layer image by using OMP with a different number of coefficients. Figure 5 shows that our approach achieves higher PSNR values than



Barbara Boat Lena Cameraman

Figure 5: Reconstruction quality between the single layer learned dictionary and our dictionary. Horizontal axis is the sparsity (T_n per patch), and the vertical axis is PSNR in dB. Red: conventional dictionary, Black: dictionary generated by our algorithm.



KSVD Ours KSVD Ours

Figure 6: Left: The frequency graphs of atoms when 15 coefficients are used in reconstruction. Right: the dictionaries generated by the KSVD and our method.

using the original KSVD.

We also notice that the probability of each dictionary atom utilized in our reconstruction is different from the KSVD dictionary. In [31] a method called Equiprobable Matching Pursuit (EMP) where a probability constraint is incorporated to prevent a few atoms dominating the reconstruction is proposed. Our nonlinear dictionary learning also generates a dictionary that can avert having one or two atoms to become dominant to others, achieving the same goal as EMP without imposing any additional constraints. Figure 6 shows that the atoms in our dictionary are utilized more uniformly. In comparison, KSVD exploits one atom more often than others. At the same time, the dictionary atoms learned by our algorithm are more diverse than the ones in the KSVD dictionary.

4.3. Layers Matter

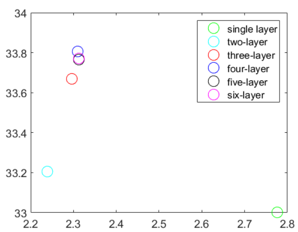
There is a positive correlation between the quality of the reconstruction and the number of layers in our cascaded framework. We also notice in the bottom graph in Fig. 7 that the computational complexity does not change much with the increase of the layers. Does this mean the deeper hierarchical models are better?

To seek an answer to the question of the optimal number of the layers, we analyze the reconstruction results for different number of layers from 1 to 6 on three test images (Boat, Barbara, Lena) as reported in Fig. 7. We observe that our multi-layer reconstruction is more accurate than single layer reconstruction while using a smaller number of coefficients. However, the results do not improve remarkably after the fourth-layer reconstruction. Since the number of patches extracted from the fifth and sixth layers are only 625 and 72, respectively, which is only approximately 1/400 and 1/4000 of the number of patches extracted from the finest layer, they hardly influence the dictionary building, leading a larger error for these two layers (as a result, using more coefficients in the following layers to fix this). On the other hand, reconstructing a 8×8 patch in the fifth layer is equal to a 128×128 patch in the finest layer, which is too large to estimate accurately using small dictionary atoms. We find that in most images, a four-layer pyramid provides an optimal hierarchical representation.

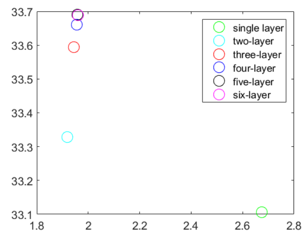
As in Fig. 7, our method does not increase the computational load in comparison to a single layer and it would benefit from faster optimization techniques for a single layer. A discussion on the converge analysis of different optimization techniques for a single layer such as K-SVD, Accelerated Plain Dictionary Learning, etc. can be found in [32].

5. Experimental Analysis

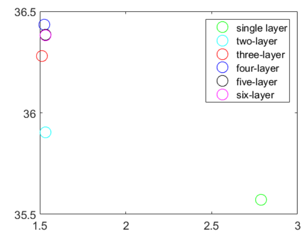
To demonstrate the flexibility of our method, we evaluate its performance on three different and popular image processing tasks: image coding, image denoising, and image inpainting. For a comprehensive evaluation, we build five different image datasets, where each dataset contains 50 images of specific object classes: animals, landscapes, textures, faces, and fingerprints (all color except the fingerprint images, which are grayscale). Some of these images are selected from the BSD300 [33] and CelebA [34] datasets, and the rest are downloaded from the websites. The size of the images in these datasets



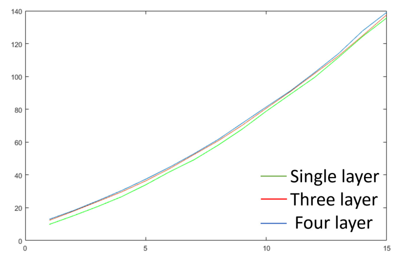
Accuracy - Boat



Accuracy - Barbara



Accuracy - Lena



Computational load

Figure 7: Top: The PSNR vs the average number of coefficients per pixel for different layer versions of our method and single-layer version. Bottom: Computational times with respect to the number of coefficients used (single-layer is KSVD, others are our cascade method).

varies from 256×256 to 480×440. The grayscale versions of sample images are shown in Fig. (8).

5.1. Image Coding

We compare our method with 5 state-of-the-art dictionary learning algorithms including both single and multi-scale methods: approximate KSVD (a-KSVD) [29], ODL [14], KSVD [2], multi-scale KSVD [18], multi-scale KSVD using wavelets (multi-wavelets) [5].

For objectiveness, we use the same number of dictionary atoms for our and all other methods. Notice that, a larger dictionary would generate a sparser representations. We employ 4× overcomplete dictionaries, i.e. $\mathbf{D} \in \mathbb{R}^{64\times 256}$ except for the multi-wavelets where the dictionary in each sub-band has as many atoms as our dictionary (in favor of the multi-wavelets). For multi-scale KSVD, the maximum dimension of dictionary atom can be 16 due to the storage issue and only 2 scales can be performed. Thus, we extracted 128 atoms at each scale.

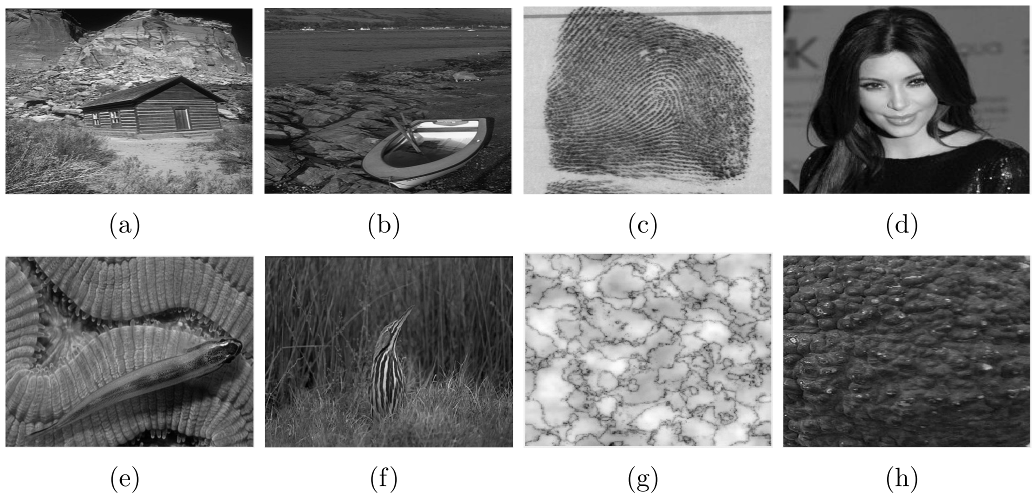
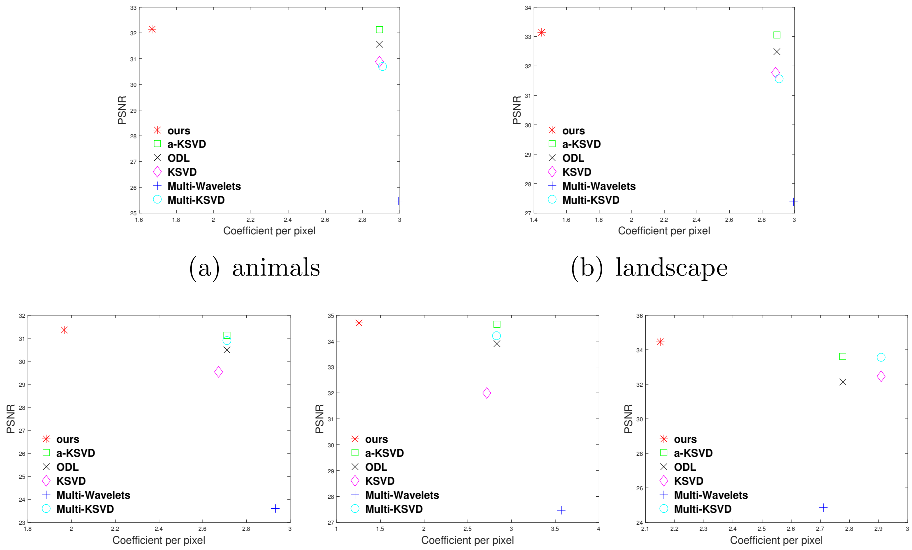


Figure 8: Sample images from 5 datasets.

Figure 9 depicts the number of coefficients per pixel as the function of the number of coefficient per each pixel. Each point is the average score per pixel for the corresponding method. As seen, our method is the best performing algorithm among the state-of-the-art. In all five image datasets, it achieves the highest PSNR scores with significantly much less number of coefficients. In these experiments, the patches are extracted by 1-pixel overlapping in all images. We use 8×8 blocks on each layer, and the cascade comprises 4 layers. Since the blocks in every layer have the same size, the lower resolution blocks efficiently represent larger receptive fields when they are upsampling onto a higher resolution.

When decoding on the coarsest resolution, our method employs $8{\times}8$ blocks, which corresponds to $8 \cdot 2^{n-1} \times 8 \cdot 2^{n-1}$ patches on the finest (original) resolution using the same dictionary atoms. Since there is a single global dictionary after the second pass, all layers share the same atoms. Even though this may resemble the quadtree structure, our method is not limited by the size of the dictionary (patch size, i.e., the dimensionality of the atoms, and the number of the atoms). Furthermore, it is as fast as the baseline single-scale dictionary learning and sparse coding methods.

Compared with other algorithms, our method can save an outstanding 55.6%, 42.23% and 49.95% coefficients for the face, animals, and landscape datasets, respectively. For the image classes where spatial texture is dominant, our method is also superior by decreasing the number of coefficient



(a) animals (b) landscape

(c) texture (d) face (e) fingerprint

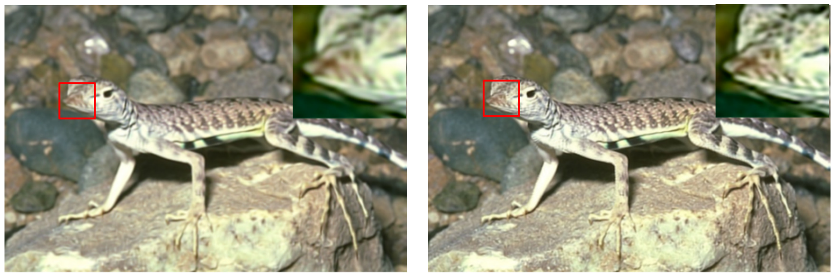
Figure 9: Reconstruction results on different 5 different image datasets. The horizontal axis represents the number of coefficient per pixel and the vertical axis is the quality in terms of PSNR (dB).

by 27.74% and 22.38% for the texture and fingerprint datasets. The ratio is defined as $(c_1 - c_0)/c_1$, where c_0 is the number of the coefficients employed by our algorithm and c_1 is the number of the coefficients used by the second best algorithm. Note that, for all the five datasets, our algorithm achieves the highest PSNR while using much fewer coefficients. The second best algorithm is a-KSVD (Fig. (9)). Sample image coding results for qualitative assessment are given in Fig. 10. As shown, a-KSVD image coding generates inferior results even though it uses more coefficients.

5.2. Image Denoising

We also analyze the image denoising performance of our method and make comparisons with five dictionary learning algorithms. We note that the state of the art in denoising use collaborative and non-local techniques such as BM3D [35] and LSSC [1]. However, our goal here is not to design a yet another collaborative scheme. Instead, we aim to understand how our method compares to other dictionary learning methods.

We minimize the cost function in Eqn. (12) for denoising. We use the



(a) a-KSVD: 28.68 db PSNR (b) Our method: 32.62 db PSNR

Figure 10: Image coding results the comparison between a-KSVD and our method. Our method uses 1309035 coefficients and achieves 32.62 db PSNR score, while a-KSVD uses 1332286 coefficients to get 28.65 dB PSNR. our method is almost **4 dB** better. Enlarged red regions are shown on the top-right corner of each image. As visible, our method produces more detailed reconstructions.

Table 1: Denoising results on different test images for $\sigma = 10$. (M-W: multi-wavelets)

	KSVD	ODL	a-KSVD	M-W	m-KSVD	Ours
a	31.10	30.98	31.05	30.95	**31.16**	30.61
b	32.93	**33.05**	32.93	32.74	33.02	32.91
c	34.05	**34.09**	34.01	33.99	33.42	**34.09**
d	35.61	35.67	35.62	32.36	35.52	**35.70**
e	34.18	**34.38**	34.20	34.13	34.07	34.33
f	34.35	**34.57**	34.38	34.51	34.47	34.52
g	33.18	33.52	33.22	33.49	33.50	**33.74**
h	33.90	33.91	**34.00**	33.85	33.85	**34.00**

Table 2: Denoising results on different test images for $\sigma = 30$.

	KSVD	ODL	a-KSVD	M-W	m-KSVD	Ours
a	25.03	25.04	25.06	25.08	**25.10**	25.03
b	27.79	27.84	27.78	27.83	27.78	**27.85**
c	27.48	26.96	27.46	**28.38**	27.77	28.01
d	30.33	**30.39**	30.35	30.11	30.13	30.29
e	28.36	28.30	28.32	**29.10**	28.53	29.08
f	28.50	28.46	28.44	**29.21**	28.59	29.06
g	27.71	27.46	27.69	28.12	27.86	**28.20**
h	28.30	28.29	28.27	28.69	28.37	**28.83**

Figure 11: Denoised images. Additive zero-mean Gaussian noise with $\sigma = 30$.

Table 3: Denoising results on test images for $\sigma = 50$.

	KSVD	ODL	a-KSVD	M-W	m-KSVD	Ours
a	22.75	20.80	22.74	**23.10**	22.85	22.88
b	25.75	24.27	25.73	**26.06**	25.63	25.95
c	24.19	22.65	24.16	**26.15**	24.66	25.92
d	27.80	25.09	27.84	27.79	27.52	**27.85**
e	26.65	26.05	26.63	27.09	26.42	**27.19**
f	26.72	25.27	26.70	**27.14**	26.43	26.85
g	26.04	25.73	26.05	**26.27**	25.80	26.19
h	26.45	25.82	26.43	**26.63**	26.20	26.56

difference between the downsampled input image and aggregated reconstructions at each layer to terminate the OMP.

$$\hat{\mathbf{x}}_n^{ij} = \arg\min_{\mathbf{x}_i} \sum_{ij} \|\mathbf{x}_n^{ij}\|_0$$
$$\text{s.t.} \|\mathbf{R}_{ij}\mathbf{Y}_n - \mathbf{D}_n\mathbf{x}_n^{ij} + \mathbf{R}_{ij}\mathbf{U}(\hat{\mathbf{Y}}_{n+1})\|_2^2 \leq C\sigma. \tag{12}$$

Above, the reconstructed residual $\hat{\mathbf{Y}}_{n+1}$ is defined as in Eqn. (11), and σ is chosen according to the variance of the noise. As before, we choose the 4-layer cascade and 8×8 patch size. The parameters of KSVD and multi-scale wavelets are set as recommenced by original authors. We fixed all hyperparameters for all test images. Since the denoising task is totally different from image coding, we do not need to force the size of dictionary to be identical for all algorithms. In multi-scale methods, the residuals in the finer layers are mostly noise, which cannot be used to learn an efficient dictionary. Therefore, we learn a dictionary for each layer per class from the clean images, which is similar to the multi-wavelets. As shown in Fig. 11 for the 320× 480 animal image and 256×256 face image, our method achieves comparable or higher PSNR scores than the state-of-the-art methods. In addition, our method can render finer details more accurately.

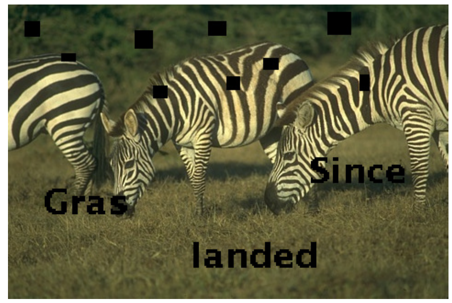
We also conducted extensive experiments with varying noise levels on a set of different types of images in Fig. 8. Table 1, 2, and 3 present the denoising results (PSNR) when the Gaussian variance is 10, 30, and 50, respectively. The leftmost columns of these tables are the corresponding ID in Fig. 8. As visible, the multi-scale wavelets perform well on images with complex textures and when the noise level is high, and ODL is suitable for lower noise levels. In comparison, our algorithm is more consistent and stable.

5.3. Image Inpainting

Image inpainting is often used for the restoration of the damaged photographs and the removal of specific artifacts such as missing pixels. Previous dictionary learning based algorithms work only when the missing area is smaller than the corresponding patch size of the dictionary atom dimensionality.

We observed that our method generates the best image inpainting results. As demonstrated in Fig. 1 our method can restore the missing image regions that are remarkably much larger than the dimension of dictionary atoms, outperforming the state-of-the-art methods. By reconstructing the

(a) original image



(b) corrupted image

PSNR 28.74

(c) KSVD

PSNR 34.88

(d) Ours

Figure 12: A sample 480×320 image from the animal dataset is corrupted with large artifacts and missing blocks. The sizes of the artifacts range from 8 to 32 pixels. Our method efficiently removes the artifacts.

Ours: 40 dB KSVD: 26 dB Ours: 36 dB KSVD: 22 dB

Figure 13: (a-b) Inpainting results for 8×8 and 14×14 missing blocks. (c-d) Results for 16×16 to 32×32 missing blocks.

Table 4: Image In-painting Results

	8	14	20	26	32	38	44	50
KSVD	34.76	26.96	22.03	20.18	18.86	16.43	16.48	14.24
Ours	41.59	40.78	37.54	33.80	30.25	25.87	26.12	23.44

image starting at the coarsest layer, we can fix completely missing regions. The larger the missing area, the smoother the restored image becomes. In comparison, single-scale based methods fail completely.

Given the mask $\mathbf{M}$ of missing pixels, our formulation in each layer is

$$\begin{aligned} \hat{\mathbf{x}}_n^{ij} &= \arg\min_{\mathbf{x}_n} \sum_{ij} \|\mathbf{R}_{ij}\mathbf{M} \otimes (\mathbf{R}_{ij}\mathbf{Y}_n^{'} - \mathbf{D}_n\mathbf{x}_n)\|_2^2 \\ \text{s.t.} \quad & \|\mathbf{x}_n^{ij}\|_0 \leq T_n \end{aligned} \tag{13}$$

where we denote $\otimes$ as the element-wise multiplication between two vectors.

Figure 12 shows that our algorithm can fill in the big holes where the KSVD fails. To analyze our algorithm further, we randomly remove 8 different sized squares (8, 14, 20, 26, 32, 38, 44, and 50) at 1 to 6 image locations each (8 to 48 holes at each try) in the given image in Fig. 13. When the missing area is small, e.g. 8×8 and 14×14, our algorithm can recover with a high PSNR of 40 dB, which is approximately 14 dB higher than the KSVD. When the missing area size is between 16×16 to 32×32, our method can still recover with 36 dB PSNR but KSVD degrades to around 22 dB. With the missing areas growing, our algorithm still outperforms the KSVD almost 10 dB. Here, we compare with the KSVD algorithm since the multi-scale KSVD simply increases the dimension of atoms, which leads proportionally more atoms to form an overcomplete dictionary. At the same time, multi-scale KSVD still fails to handle holes larger than the dimensionality of the atoms.

6. Conclusion

We presented a non-linear dictionary learning and sparse coding method on cascaded residuals. Our cascade allows capturing both local and global information. Its coarse-to-fine structure prevent from reconstructing the regions that can be well represented by the coarser layers. Our sparse coding can be used to progressively improve the quality of the decoded image.

Our method provides significant improvement over the state-of-the-art solutions in terms of the quality of reconstructed image, reduction in the number of coefficients, and computational complexity. It generates much higher quality images using less number of coefficients. It produces superior results on image inpainting, in particular, in handling of very large ratios of missing pixels and large gaps.

Acknowledgment

This work was supported by the Australian Research Council's Discovery Projects funding scheme (project DP150104645).

7. Reference

[1] J. Mairal, F. Bach, J. Ponce, G. Sapiro, A. Zisserman, Non-local sparse models for image restoration, in: Computer Vision, 2009 IEEE 12th International Conference on, IEEE, pp. 2272–2279.

[2] M. Aharon, M. Elad, A. Bruckstein, K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation, IEEE Transactions on Signal Processing 54 (2006) 4311–4322.

[3] J. Mairal, J. Ponce, G. Sapiro, A. Zisserman, F. R. Bach, Supervised dictionary learning, in: Advances in neural information processing systems, pp. 1033–1040.

[4] R. Yan, L. Shao, Y. Liu, Nonlocal hierarchical dictionary learning using wavelets for image denoising, Image Processing, IEEE Transactions on 22 (2013) 4689–4698.

[5] B. Ophir, M. Lustig, M. Elad, Multi-Scale Dictionary Learning Using Wavelets, IEEE Journal of Selected Topics in Signal Processing 5 (2011) 1014–1024.

[6] J. Sulam, B. Ophir, M. Elad, Image denoising through multi-scale learnt dictionaries, in: Image Processing (ICIP), 2014 IEEE International Conference on, IEEE, pp. 808–812.

[7] N. Ahmed, T. Natarajan, K. R. Rao, Discrete cosine transfom, IEEE transactions on Computers (1974) 90–93.

[8] S. Mallat, A wavelet tour of signal processing, Academic press, 1999.

[9] E. J. Candes, D. L. Donoho, Curvelets: A surprisingly effective non-adaptive representation for objects with edges, Technical Report, DTIC Document, 2000.

[10] M. N. Do, M. Vetterli, The contourlet transform: an efficient directional multiresolution image representation, Image Processing, IEEE Transactions on 14 (2005) 2091–2106.

[11] D. Labate, W.-Q. Lim, G. Kutyniok, G. Weiss, Sparse multidimensional representation using shearlets, in: Optics & Photonics 2005, International Society for Optics and Photonics, pp. 59140U–59140U.

[12] K. Engan, S. O. Aase, J. H. Husoy, Method of optimal directions for frame design, 1999 IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings. ICASSP99 5 (1999) 2443–2446.

[13] R. Vidal, Y. Ma, S. Sastry, Generalized principal component analysis (gpca), Pattern Analysis and Machine Intelligence, IEEE Transactions on 27 (2005) 1945–1959.

[14] J. Mairal, F. Bach, J. Ponce, G. Sapiro, Online dictionary learning for sparse coding, Proceedings of the 26th International Conference on Machine Learning (2009) 1–8.

[15] J. Tarquino, A. Rueda, E. Romero, A multiscalesparse representation for diffusion weighted imaging (dwi) super-resolution, in: Biomedical Imaging (ISBI), 2014 IEEE 11th International Symposium on, IEEE, pp. 983–986.

[16] Y. Liu, S. Liu, Z. Wang, A general framework for image fusion based on multi-scale transform and sparse representation, Information Fusion 24 (2015) 147–164.

[17] H. Yin, Sparse representation with learned multiscale dictionary for image fusion, Neurocomputing 148 (2015) 600–610.

[18] J. Mairal, G. Sapiro, M. Elad, Learning multiscale sparse representations for image and video restoration, Multiscale Modeling & Simulation 7 (2008) 214–241.

[19] S. G. Mallat, Z. Zhang, Matching pursuits with time-frequency dictionaries, IEEE Transactions on Signal Processing 41 (1993) 3397–3415.

[20] Y. C. Pati, R. Rezaiifar, P. Krishnaprasad, Orthogonal matching pursuit: Recursive function approximation with applications to wavelet decomposition (1993) 40–44.

[21] S. S. Chen, D. L. Donoho, M. A. Saunders, Atomic decomposition by basis pursuit, SIAM review 43 (2001) 129–159.

[22] I. F. Gorodnitsky, B. D. Rao, Sparse signal reconstruction from limited data using FOCUSS: A re-weighted minimum norm algorithm, IEEE Transactions on Signal Processing 45 (1997) 600–616.

[23] B. Efron, T. Hastie, I. Johnstone, R. Tibshirani, et al., Least angle regression, The Annals of statistics 32 (2004) 407–499.

[24] E. Le Pennec, S. Mallat, Sparse geometric image representations with bandelets, IEEE Transactions on Image Processing 14 (2005) 423–438.

[25] S. G. Mallat, A theory for multiresolution signal decomposition: the wavelet representation, Pattern Analysis and Machine Intelligence, IEEE Transactions on 11 (1989) 674–693.

[26] E. P. Simoncelli, W. T. Freeman, The steerable pyramid: A flexible architecture for multi-scale derivative computation, in: icip, IEEE, p. 3444.

[27] P. J. Burt, E. H. Adelson, The Laplacian Pyramid as a Compact Image Code, IEEE Transactions on Communications 31 (1983) 532–540.

[28] W. Dong, L. Zhang, R. Lukac, G. Shi, Sparse representation based image interpolation with nonlocal autoregressive modeling, Image Processing, IEEE Transactions on 22 (2013) 1382–1394.

[29] R. Rubinstein, M. Zibulevsky, M. Elad, Efficient implementation of the k-svd algorithm using batch orthogonal matching pursuit, CS Technion 40 (2008) 1–15.

[30] J. A. Tropp, Greed is good: Algorithmic results for sparse approximation, Information Theory, IEEE Transactions on 50 (2004) 2231–2242.

[31] F. Sandin, S. Martin-del Campo, Dictionary learning with equiprobable matching pursuit, arXiv preprint arXiv:1611.09333 (2016).

[32] C. Bao, H. Ji, Y. Quan, Z. Shen, Dictionary learning for sparse coding: Algorithms and convergence analysis, IEEE transactions on pattern analysis and machine intelligence 38 (2016) 1356–1369.

[33] D. Martin, C. Fowlkes, D. Tal, J. Malik, A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics, in: Proc. 8th Int'l Conf. Computer Vision, volume 2, pp. 416–423.

[34] Z. Liu, P. Luo, X. Wang, X. Tang, Deep learning face attributes in the wild, in: Proceedings of International Conference on Computer Vision (ICCV).

[35] K. Dabov, A. Foi, V. Katkovnik, K. Egiazarian, Image denoising by sparse 3-d transform-domain collaborative filtering, Image Processing, IEEE Transactions on 16 (2007) 2080–2095.