

# Sparse Coding on Cascaded Residuals

Tong Zhang<sup>1(✉)</sup> and Fatih Porikli<sup>1,2</sup>

<sup>1</sup> Australian National University, Canberra, Australia

{tong.zhang, fatih.porikli}@anu.edu.au

<sup>2</sup> Data61/CSIRO, Eveleigh, Australia

**Abstract.** This paper seeks to combine dictionary learning and hierarchical image representation in a principled way. To make dictionary atoms capturing 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. The cascade allows collaborative reconstructions at different resolutions using the same dimensional dictionary atoms. Our jointly learned dictionary comprises atoms that adapt to the information available at the coarsest layer where the support of atoms reaches their maximum range and the residual images where the supplementary details progressively refine the 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 more flexible and accurate representations using much less number of coefficients. Its computational efficiency stems from encoding at the coarsest resolution, which is minuscule, and encoding the residuals, which are relatively much sparse. Our extensive experiments on multiple datasets demonstrate that this new method is powerful in image coding, denoising, inpainting and artifact removal tasks outperforming the state-of-the-art techniques.

## 1 Introduction

Sparse representation promises noise resilience by assigning representation coefficients from dictionary atoms characterizing the clean data distribution, improved classification performance by attaining discriminative features, robustness by preventing the model from overfitting data, and semantic interpretation by allowing atoms to associate with meaningful attributes. Computer vision applications include image compression, regularization in reverse problems, feature extraction, recognition, interpolation for incomplete data, and more [1–6].

An overcomplete dictionary that leads to sparse representations can either be chosen from a predetermined set of functions or designed by adapting its content to fit a given set of samples. The performance of the predetermined dictionaries, such as overcomplete Discrete Cosine Transform (DCT) [7], wavelets [8], curvelets [9], contourlets [10], shearlets [11] and other analytic forms, depends on how suitable they are to sparsely describe the samples in question. On the other hand, the learned dictionaries are data driven and tailored for distinct

applications. Noteworthy algorithms of this type include the Method of Optimal Directions (MOD) [12], generalized PCA [13], KSVD [2], Online Dictionary Learning (ODL) [4, 14]. The learned dictionaries adapt better compared to analytic ones and provide improved performance.

In general, image based dictionary learning and sparse encoding tasks are formulated as an optimization problem

$$\arg \min_{\mathbf{D}, \mathbf{X}} \|\mathbf{Y} - \mathbf{DX}\|_F^2 \quad \text{s.t. } \|\mathbf{x}_i\|_0 \leq T, \quad (1)$$

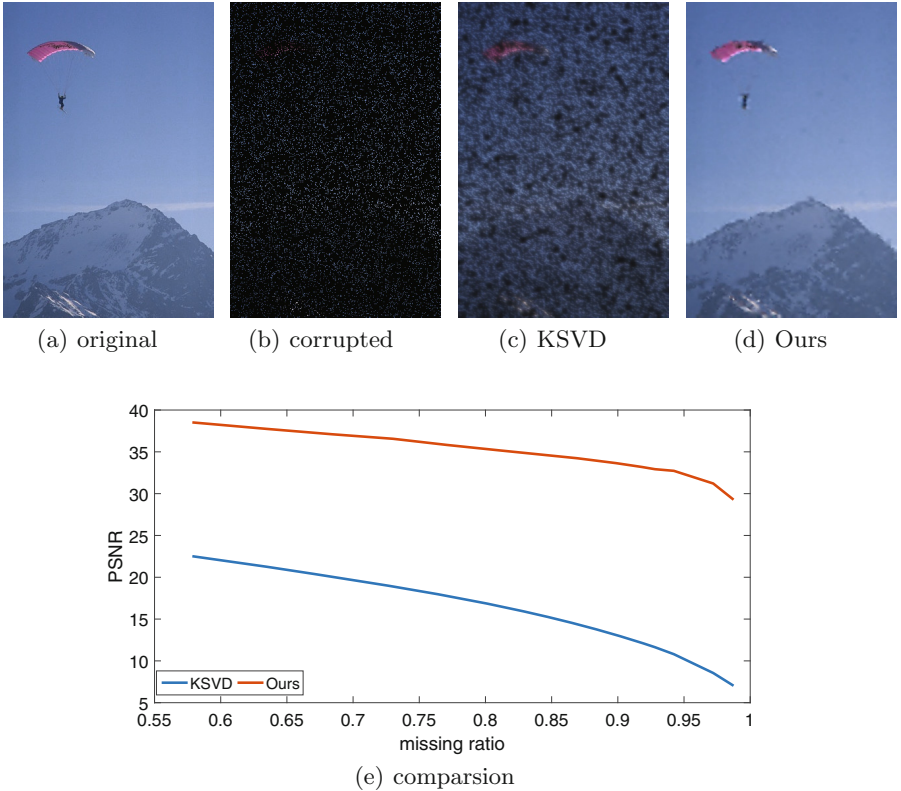
or its equivalent form

$$\arg \min_{\mathbf{X}, \mathbf{D}} \sum_i \|\mathbf{x}_i\|_0 \quad \text{s.t. } \|\mathbf{Y} - \mathbf{DX}\|_F^2 \leq \epsilon \quad (2)$$

where  $\mathbf{Y} \in \mathbb{R}^{n \times k}$  is  $k$  image patches with dimension  $n$ ,  $\mathbf{X} \in \mathbb{R}^{m \times k}$  denotes the coefficients of corresponding images,  $\mathbf{D} \in \mathbb{R}^{n \times m}$  is an overcomplete matrix,  $T$  is the number of coefficient used to describe the images, and  $\epsilon$  is the error tolerance such that once the reconstruction error is smaller than the tolerance the pursuit will be terminated. The sparsity is achieved because  $n \ll m$  and  $T \ll m$ . For an extended discussion on the solutions of above objectives, see Sect. 2.

For mathematical convenience, dictionary learning methods often employ in uniform spaces, e.g. in the vector space of  $8 \times 8$  image patches. In other words, same scale blocks are pulled from overlapping or non-overlapping image patches on a dense grid and a single-scale dictionary is learned. However, dictionary atoms learned in this fashion tend to be myopic and blind to global context since such fixed-scale patches only contain local information within their small support. Simply increasing the patch size results in adverse outcomes, i.e. decreased the flexibility of the dictionary to fit data and increased computational complexity. Moreover, optimal patch size varies depending on the underlying texture information. For example, finer partitioning by smaller blocks is preferable for textured regions, yet larger blocks would suit better for smooth areas. Suppose the image to be encoded is a  $256 \times 256$  flat (e.g. all pixels have the same value) image. Using the conventional  $8 \times 8$  overlapping blocks would require more than 60 K coefficients, yet the same image can be represented using a small number of coefficients of larger patches, even only a single coefficient in the ideal case of the patch has the size of the image.

As an alternative, multi-scale methods aim to learn dictionaries at different image resolutions for the same patch size using shearlets, wavelets, and Laplacian pyramid [4, 5, 15–17]. A major drawback of these methods is that each layer in the pyramid is either processed independently or in small frequency bands; thus reconstruction errors of coarser layers are projected directly on the finest layer. Such errors cannot be compensated by other layers. This implies, to attain a satisfactory quality, all layers need to be constructed accurately. Instead of learning in different image resolutions, [18] first builds separate dictionaries for quadtree partitioned patches and then zero-pad smaller patches to the largest scale. However, the size of the dictionary learned in this fashion is proportional to the



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

maximum patch size, which prohibits its applicability due to heavy computational load and inflated memory requirements.

Existing multi-scale dictionary learning methods overlook the redundancy between the layers. As a consequence, larger dictionaries are required, and a high number of coefficients are spent unnecessarily on smooth areas. To the best of our knowledge, no method offers a systematic solution where encodings of the coarser scales progressively enhance the reconstructions of the finer layers.

### Our Contributions

Aiming to address the above shortcomings and allow dictionary atoms to access larger context for improved descriptive capacity, here we present a computationally efficient cascade framework that employs multi-resolution residual maps for dictionary learning and sparse coding.

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, which is tiny, and encoding the residuals, which are relatively much 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 layer. Thus, the atoms in the dictionary have the same dimension still 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 supplementary details progressively refine our reconstruction objective. This allows our method to generate a flexible image representation using much less number of coefficients.

Our extensive experiments on large datasets demonstrate that this new method is powerful in image coding, denoising, inpainting and artifact removal tasks outperforming the state-of-the-art techniques. Figure 1 shows a sample inpainting result from our method where 93% of pixels are missing. As visible, by taking the take advantage of the multi-resolution cascade, we can recover even the very large missing areas.

## 2 Related Work

The nature of dictionary learning objective makes it an NP-hard problem since neither the dictionary nor the coefficients are known. To handle this, most dictionary learning algorithms alternate between 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 using as Moore-Penrose pseudo-inverse; KSVD incorporates k-means clustering and singular value decomposition by refining coefficients and dictionary atoms recursively; 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  $\ell_0$  norm. It is often approximated by greedy schemes such as Matching pursuit (MP) [19] and Orthogonal Matching Pursuit (OMP) [20]. Another popular solution is to replace the  $\ell^0$ -norm with an  $\ell^p$ -norm with  $p \leq 1$ . When  $p = 1$ , the solution can be approximated by Basis Pursuit [21], FoCUSS [22], and Least Angle Regression (LARS) [23] to count a few.

Multi-scale methods for coding have been widely studied in the past. Wavelets have become the premier multi-scale analysis tool in signal processing and many wavelets alike methods such as bandlets [24], contourlets [10], curvelets [9] as well as decomposition methods including wavelet pyramid [25], steerable pyramid [26], and Laplacian pyramid [27] have emerged. These methods aim to improve upon the pure spatial frequency analysis of Fourier transform by providing resolution in both spatial frequency and spatial location.

Nevertheless, there have been few attempts to learn multi-scale dictionaries. In [18], use of a quadtree structure was proposed. Dictionaries with different atom dimensions are learned 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 does not take the advantage of the coarse-scale information that may be more suitable to represent patches using atoms of the same size.

In order to overcome computational issues, Ophir *et al.* [5] learned 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. However, separate dictionaries are learned for the directional sub-bands, which tends to generate inferior performance when compared to single-scale KSVD in denosing task. Their following work [6] exploited multi-scale analysis and single-scale dictionary learning, and merged both outputs by weighted joint sparse coding. Since the fused dictionary is several times larger than the single-scale version, the computational complexity is high. Besides, the denoising performance is sensitive to the size and category of images. A similar work [4] built multi-resolution dictionaries on wavelet pyramid by employing k-means clustering before the ODL step. For each resolution, it clusters the patches of three sub-bands, and concatenates all dictionary atoms. Even though denoising performance improves due to non-local clustering on sub-bands, each layer requires a large dictionary, which reflects on the computationally load.

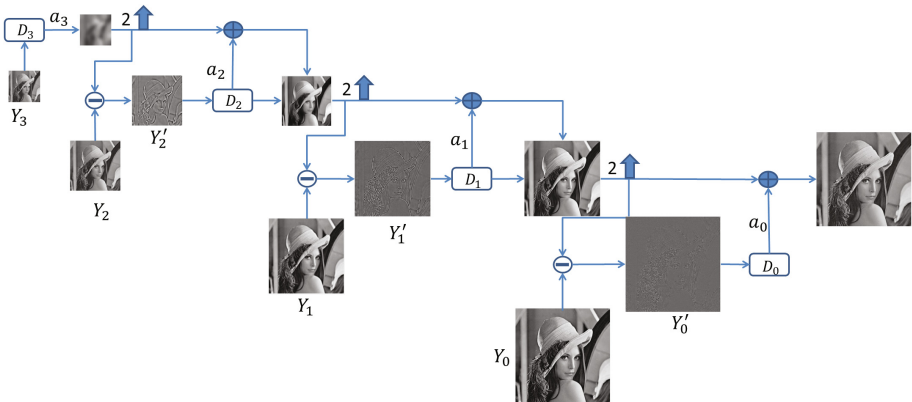
Multi-resolution sparse representations are also employed for image fusion and super-resolution. Liu *et al.* [16] fused two images by obtaining sparse coefficients for high-pass and low-pass frequency bands by OMP given the pre-trained dictionary. The fused coefficient columns in each band are chosen by maximal  $\ell_1$  norm of corresponding coefficients. Towards the same goal, Yin *et al.* [17] merged two coefficient vectors, however, the fused coefficient columns are selected by  $\ell_2$  norm. Instead of training sub-dictionaries independently, they learn  $3S + 1$  sub-dictionaries jointly, which means the dimension of the matrix is  $(3S + 1)n \times k$ , thus the learning stage is computationally expensive. In [15] proposed a multi-scale approach to super-resolve diffusion weighted images where the low-resolution dictionary is based on the shearlet transform and the high-resolution one is based on intensity. In [28] sparse representation was 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 sparse represented with

a specific dictionary. In order 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

A flow diagram of our framework is shown in Fig. 2 for a sample 4-layer cascade. Given an image  $\mathbf{Y}$ , we first construct an image pyramid  $\mathbf{Y} = \{\mathbf{Y}_0, \mathbf{Y}_1, \dots, \mathbf{Y}_N\}$  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 are Gaussian pyramid, Laplacian pyramid, bilinear interpolation, and subsampling. Images resampled with bicubic interpolation are smoother and have fewer interpolation artifacts. In contrast, bicubic interpolation considers larger support.

We employ a two-pass scheme wherein 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 from the dictionaries generated in the first-pass.



**Fig. 2.** First-pass of our method for a 4-layer cascade.  $\mathbf{Y}_0$  is the original image,  $\{\mathbf{Y}_3, \dots, \mathbf{Y}_0\}$  denote each layer of the image  $\mathbf{Y}_3$  pyramid, and  $\{\mathbf{D}_3, \dots, \mathbf{D}_0\}$  are the dictionaries.  $\mathbf{D}_3$  is learned from the downsampled image, remaining dictionaries are learned from the residuals  $\{\mathbf{Y}'_2, \mathbf{Y}'_1, \mathbf{Y}'_0\}$ .  $\alpha_n$  are the coefficients used to reconstruct each layer.

#### First-pass

We start at the coarsest layer  $N$  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 use same size  $b \times b$  patches. A patch in the layer  $n$  corresponds

---

**Algorithm 1.** Cascade sparse coding
 

---

**Input:**

- 1:  $N$  (the highest pyramid layer),  $\mathbf{Y}$  (image),
- 2:  $T_n$  (number of coefficient used in layer  $n$ )

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

- 3:  $\mathbf{Y}_n \leftarrow$  subsampling( $\mathbf{Y}$ ,  $2^n$ )
  - 4: **for**  $n = \{N, N-1, \dots, 0\}$  **do**
  - 5:     **if**  $n = N$  **then**
  - 6:          $\mathbf{Y}'_n \leftarrow \mathbf{Y}_n$
  - 7:     **else**
  - 8:          $\mathbf{Y}'_n \leftarrow \mathbf{Y}_n -$  upsample( $\hat{\mathbf{Y}}_{n+1}$ , 2)
  - 9:     Perform KSVD to learn dictionary  $\hat{\mathbf{D}}_n$  and encode  $\mathbf{Y}'_n$
  - 10:      $\forall ij \{ \hat{\mathbf{x}}_n^{ij}, \hat{\mathbf{D}}_n \} \leftarrow \arg \min_{\mathbf{x}_n^{ij}, \mathbf{D}_n} \sum_{ij} \|\mathbf{R}_{ij} \mathbf{Y}'_n - \mathbf{D}_n \mathbf{x}_n^{ij}\|_2^2 \quad \text{s.t. } \|\mathbf{x}_n^{ij}\|_0 \leq T_n$
  - 11:     **if**  $n = N$  **then**
  - 12:          $\hat{\mathbf{Y}}_n \leftarrow (\sum_{ij} \mathbf{R}_{ij}^T \mathbf{R}_{ij})^{-1} (\sum_{ij} \mathbf{R}_{ij}^T \hat{\mathbf{D}}_n \hat{\mathbf{x}}_n^{ij})$
  - 13:     **else**
  - 14:          $\hat{\mathbf{Y}}_n \leftarrow (\sum_{ij} \mathbf{R}_{ij}^T \mathbf{R}_{ij})^{-1} (\sum_{ij} \mathbf{R}_{ij}^T \hat{\mathbf{D}}_n \hat{\mathbf{x}}_n^{ij}) +$  upsample( $\hat{\mathbf{Y}}_{n+1}$ , 2)
  - 15:  $\mathbf{Y}' \leftarrow \{\mathbf{Y}'_N, \mathbf{Y}'_{N-1}, \dots, \mathbf{Y}'_0\}$
  - 16:  $\forall ij \hat{\mathbf{D}}_{global} \leftarrow \arg \min_{\mathbf{D}} \sum_{ij} \|\mathbf{R}_{ij} \mathbf{Y}' - \mathbf{D} \mathbf{x}^{ij}\|_2^2 \quad \text{s.t. } \|\mathbf{x}^{ij}\|_0 \leq T$
  - 17: **Reconstruction:**
  - 18:  $\hat{\mathbf{Y}} \leftarrow 0$
  - 19: **for**  $n = \{N, N-1, \dots, 0\}$  **do**
  - 20:      $\mathbf{Y}'_n = \mathbf{Y}'_n -$  upsample( $\hat{\mathbf{Y}}$ , 2)
  - 21:      $\forall ij \{ \hat{\mathbf{x}}_n^{ij} \} \leftarrow \arg \min_{\mathbf{x}_n^{ij}} \sum_{ij} \|\mathbf{R}_{ij} \mathbf{Y}'_n - \hat{\mathbf{D}}_{global} \mathbf{x}_n^{ij}\|_2^2 \quad \text{s.t. } \|\mathbf{x}_n^{ij}\|_0 \leq T_n$
  - 22:      $\hat{\mathbf{Y}} \leftarrow (\sum_{ij} \mathbf{R}_{ij}^T \mathbf{R}_{ij})^{-1} (\sum_{ij} \mathbf{R}_{ij}^T \hat{\mathbf{D}}_{global} \hat{\mathbf{x}}_n^{ij}) +$  upsample( $\hat{\mathbf{Y}}$ , 2)
  - 23: **return**
- 

to a  $(b \times 2^n) \times (b \times 2^n)$  area in the original image. Algorithm 1 summarizes the first-pass.

**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

$$\arg \min_{\mathbf{D}_N, \mathbf{x}_N^{ij}} \sum_{ij} \|\mathbf{R}_{ij} \mathbf{Y}_N - \mathbf{D}_N \mathbf{x}_N^{ij}\|_2^2 + \lambda \|\mathbf{x}_N^{ij}\|_0 \quad (3)$$

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 in a column vector form. The parameter  $\lambda$  balances the data fidelity term and the regularization term, and  $\mathbf{x}_N^{ij}$  denotes the coefficients for the patch  $(i, j)$ .

We initialize the dictionary  $\mathbf{D}_N$  with a DCT basis by extracting several atoms from the DCT bases and applying Kronecker product. It is possible to choose any dictionary update methods such as KSVD [2], approximate KSVD [29],

MOD [12], and ODL [14]. Both ODL and approximate KSVD can achieve the same PSNR with less coefficients. In order to reveal the strength of our method, we choose the original KSVD to update the dictionaries. Therefore, we do a sequence of rank-one approximations that update both the dictionary atoms and the coefficients.

Iteratively, we first fix all coefficients  $\hat{\mathbf{x}}_N^{ij}$  and select each dictionary atom one by one  $\mathbf{d}_N^l$ ,  $l = \{1, 2, \dots, k\}$ . For any atom  $\mathbf{d}_N^l$ , we extract the patches, which are composed by the atom  $(i, j) \in \mathbf{d}_N^l$ , to compute its residual. The corresponding coefficients are denoted as  $\mathbf{x}_N^{ij}(l)$ , which are the non-zero entries of the  $l$ -th row of coefficient matrix

$$\mathbf{e}_N^{ij}(l) = \mathbf{R}_{ij} \mathbf{Y}_N - \hat{\mathbf{D}}_N \mathbf{x}_N^{ij} + \mathbf{d}_N^l \mathbf{x}_N^{ij}(l). \quad (4)$$

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

$$\{\hat{\mathbf{d}}_N^l, \hat{\mathbf{x}}_N(l)\} = \arg \min_{\mathbf{d}, \mathbf{x}} \|\mathbf{E}_N^l - \mathbf{d}\mathbf{x}\|_F^2. \quad (5)$$

Finally, we perform 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 the first column of matrix  $\Sigma(1, 1)\mathbf{V}$ . In every iteration all dictionary atoms and coefficients are updated simultaneously.

**Sparse coding:** After getting the updated dictionary, sparse coding is done with the Orthogonal Matching Pursuit (OMP), a greedy algorithm that is computationally efficient [30]. The sparse coding stops when the number of coefficient reaches the upper limit  $T_N$  or the reconstruction error becomes less than threshold

$$\hat{\mathbf{x}}_N^{ij} = \arg \min_{\mathbf{x}_N^{ij}} \sum_{ij} \|\mathbf{R}_{ij} \mathbf{Y}'_N - \hat{\mathbf{D}}_n \mathbf{x}_n^{ij}\|_2^2 \quad \text{s.t.} \quad \|\mathbf{x}_n^{ij}\|_0 \leq T_N. \quad (6)$$

Putting the updated coefficient matrix  $\hat{\mathbf{x}}_N^{ij}$  back into Dictionary learning to update the dictionary and coefficient until reaching the iteration times.

**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 to emerge. Larger values of  $T_n$  generates a more accurate reconstructed image. Thus, the total energy of residuals will diminish. 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 wide 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

$$\hat{\mathbf{Y}}_N = \left( \sum_{ij} \mathbf{R}_{ij}^T \mathbf{R}_{ij} \right)^{-1} \left( \sum_{ij} \mathbf{R}_{ij}^T \hat{\mathbf{D}}_N \hat{\mathbf{x}}_N^{ij} \right). \quad (7)$$



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. As the procedure of dictionary learning and sparse coding in the  $N$ -th layer, we reconstruct residual  $\hat{\mathbf{Y}}'_{N-1}$  by training a residual dictionary  $\mathbf{D}_{N-1}$  from the residual image itself. We keep encoding and decoding on residuals up to the finest layer. The cascade residual dictionary learning and reconstruction can be expressed as follows:

$$\{\hat{\mathbf{x}}_n^{ij}, \hat{\mathbf{D}}_n\} = \arg \min_{\mathbf{x}_n^{ij}, \mathbf{D}_n} \sum_{ij} \|\mathbf{R}_{ij} \mathbf{Y}'_n - \mathbf{D}_n \mathbf{x}_n^{ij}\|_2^2 \quad \text{s.t.} \quad \|\mathbf{x}_n^{ij}\|_0 \leq T_n, \quad (8)$$

where residual image is

$$\mathbf{Y}'_n = \begin{cases} \mathbf{Y}_n - \mathbf{U}(\hat{\mathbf{Y}}_{n+1}), & 0 \leq n < N \\ \mathbf{Y}_N, & n = N, \end{cases} \quad (9)$$

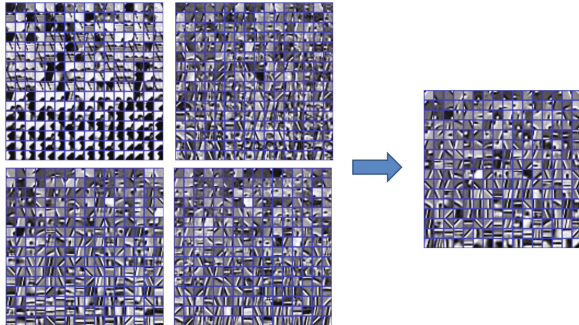
and the reconstructed residual is

$$\hat{\mathbf{Y}}_n = \begin{cases} (\sum_{ij} \mathbf{R}_{ij}^T \mathbf{R}_{ij})^{-1} (\sum_{ij} \mathbf{R}_{ij}^T \hat{\mathbf{D}}_n \hat{\mathbf{x}}_n^{ij}) + \mathbf{U}(\hat{\mathbf{Y}}_{n+1}), & 0 \leq n < N \\ (\sum_{ij} \mathbf{R}_{ij}^T \mathbf{R}_{ij})^{-1} (\sum_{ij} \mathbf{R}_{ij}^T \hat{\mathbf{D}}_n \hat{\mathbf{x}}_n^{ij}) & n = N. \end{cases} \quad (10)$$

The more coefficients used, which reduce the error caused by sparse representation. Since we are pursuing sparser representation, less number of coefficient would be better.

### 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



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

dictionary atoms. However, it is not straightforward to determine the optimal number of atoms for each dictionary since finer level residuals depend on coarser ones. Rather than keeping all dictionaries, we train a global dictionary  $\mathbf{D}$  using patches from  $\mathbf{Y}' = \{\mathbf{Y}_N, \mathbf{Y}'_{N-1}, \dots, \mathbf{Y}'_0\}$ . As illustrated in Fig. 3, the dictionaries learned from  $\mathbf{Y}'$  in the first pass are redundant. The overall dictionary is less repetitive and more general to reconstruct all four layers. This allows us to select most useful atoms automatically without making sub-optimal layer-wise decisions. Notice that, in this procedure the number of coefficient can be arbitrarily chosen depending on the target quality of each layer.

## 4 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. Our method is shown to generate the best image inpainting results and provide the most compact set of coding coefficients.

### 4.1 Image Coding

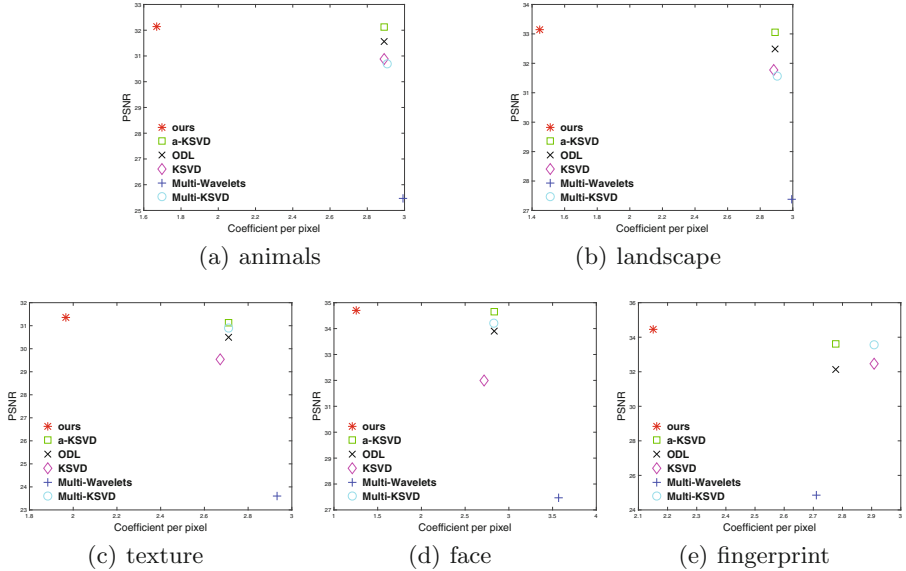
We compare our method with five state-of-the-art dictionary learning algorithms including both single and multi-scale methods: approximate KSVD (a-KSVD) [29], ODL [14], KSVD [2] Multi-scale KSVD [18], Multi-scale KSVD using wavelets (Multi-wavelets) [5].

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-times over-complete 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 Multi-wavelets).

For a comprehensive evaluation, we build five different image datasets, each contains 50 samples of a specific class of images: animals, landscape, texture, face, and fingerprint.

Figure 4 depicts the number of coefficients per pixel vs. PSNR as the function of number of coefficient per each pixel. Each point is the average score 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 higher 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  $8 \times 8$  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.

From another perspective, when decoding on the coarsest resolution, our method first employs  $8 \times 8$  blocks, which corresponds to  $8 * 2^{n-1} \times 8 * 2^{n-1}$  region on the finest resolution using the same dictionary atoms. Since there is a single global dictionary after the second pass, all layers share the same atoms. This resembles the quadtree structure, however, our method is not limited by the



**Fig. 4.** 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).

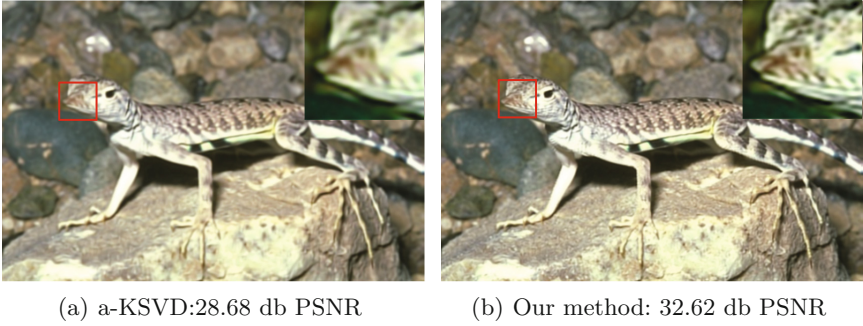
size of the dictionary (dimension of patches - atoms - and number of atoms) and it is as fast as single-scale dictionary learning and sparse coding. For Multiscale KSVD, the maximum dimension of dictionary atom can be 8 and only 2 scales can be performed. Thus, we extracted 128 atoms at each scale.

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 by 27.74% and 22.38% for the texture and fingerprint datasets. Sample image coding results for qualitative assessment are given in Fig. 5. As shown, a-KSVD image coding is inferior to our even though a-KSVD uses more coefficients.

## 4.2 Image Denoising

We also analyze the image denoising performance of our method. We make comparison with five dictionary learning algorithms. We note that the state-of-the-art is collaborative and non-local techniques, such as BM3D [31], LSSC [1], yet we do not engineer a collaborative scheme. Our goal here is to understand how our method compares to other dictionary learning methods.

We minimize the cost function in Eq. (11) for denoising. We use the difference between the downsampled input image and aggregated reconstructions at each layer to terminate the OMP.



**Fig. 5.** 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 accurate reconstructions.



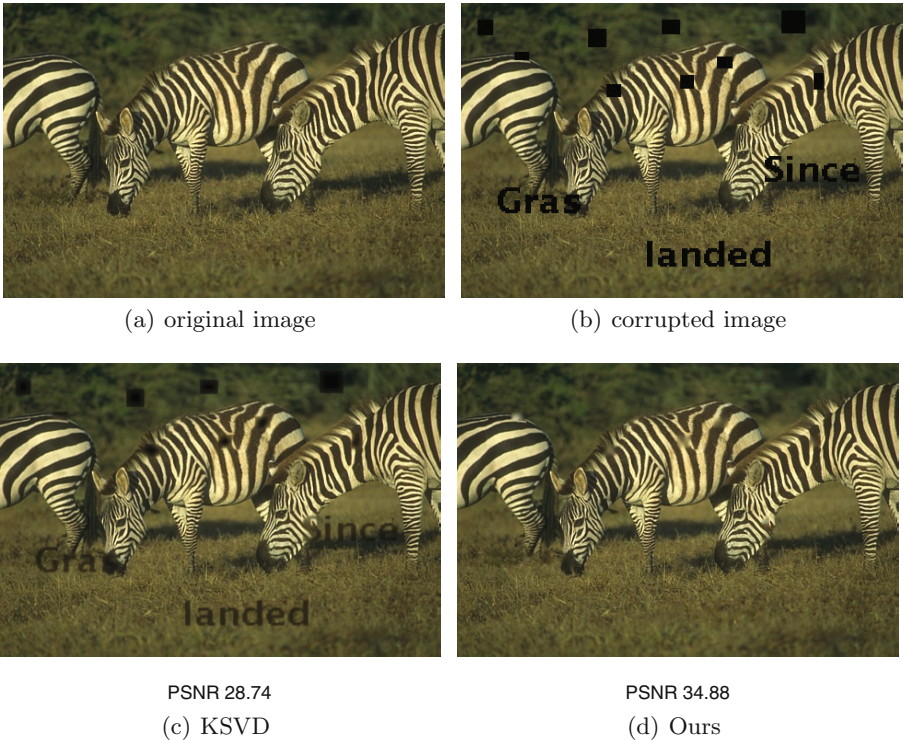
**Fig. 6.** Denoised face images. Additive zero-mean Gaussian noise with  $\sigma = 30$ .

$$\begin{aligned} & \arg \min_{\mathbf{x}_i} \sum_{ij} \|\mathbf{x}_n^{ij}\|_0 \\ & \text{s.t.} \|\mathbf{R}_{ij} \mathbf{Y}_n - \mathbf{D}_n \mathbf{x}_n^{ij} + \mathbf{R}_{ij} \mathbf{U}(\hat{\mathbf{Y}}_{n+1})\|_2^2 \leq C\sigma \end{aligned} \quad (11)$$

Above, the reconstructed residual  $\hat{\mathbf{Y}}_{n+1}$  is defined as in Eq. (10), and  $\sigma$  is chosen according to the variance of the noise. As before, we choose the 4-layer cascade and  $8 \times 8$  patch size. The parameters of KSVD and multi-scale wavelets are set as recommended by original authors. We fixed all parameters for all test images. As shown in Fig. 6, our method achieves higher PSNR scores than the state-of-the-art. In addition, it can render finer details more accurately.

### 4.3 Image Inpainting

Image inpainting is often used for restoration of damaged photographs and removal of specific artifacts such as missing pixels. Previous dictionary learning based algorithms work when the missing area is small and smaller than the dimension of dictionary atoms.



**Fig. 7.** The original image is corrupted with large artifacts. The sizes of the artifacts range from 8 to 32 pixels. Our method efficiently removes the artifacts.



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 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

$$\hat{\mathbf{x}}_n^{ij} = \arg \min_{\mathbf{x}_n} \sum_{ij} \|\mathbf{R}_{ij}\mathbf{M} \otimes (\mathbf{R}_{ij}\mathbf{Y}'_n - \mathbf{D}_n\mathbf{x}_n)\|_2^2 \quad (12)$$

subject to  $\|\mathbf{x}_n^{ij}\|_0 \leq T_n$

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

Figure 7 shows that our algorithm can fix big holes and gaps but the KSVD can not. In this experiments we only compare with KSVD algorithm, because multiscale KSVD simply increases the dimension of atoms, which leads proportionally more atoms to form an overcomplete dictionary. At the same time, multiscale KSVD still fails to handle holes larger than the dimension of atoms.

## 5 Conclusion

We presented a 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).

## References

1. Mairal, J., Bach, F., Ponce, J., Sapiro, G., Zisserman, A.: Non-local sparse models for image restoration. In: 2009 IEEE 12th International Conference on Computer Vision, pp. 2272–2279. IEEE (2009)
2. Aharon, M., Elad, M., Bruckstein, A.: K-SVD: an algorithm for designing overcomplete dictionaries for sparse representation. *IEEE Trans. Sig. Process.* **54**, 4311–4322 (2006)

3. Mairal, J., Ponce, J., Sapiro, G., Zisserman, A., Bach, F.R.: Supervised dictionary learning. In: *Advances in Neural Information Processing Systems*, pp. 1033–1040 (2009)
4. Yan, R., Shao, L., Liu, Y.: Nonlocal hierarchical dictionary learning using wavelets for image denoising. *IEEE Trans. Image Process.* **22**, 4689–4698 (2013)
5. Ophir, B., Lustig, M., Elad, M.: Multi-scale dictionary learning using wavelets. *IEEE J. Sel. Top. Sig. Process.* **5**, 1014–1024 (2011)
6. Sulam, J., Ophir, B., Elad, M.: Image denoising through multi-scale learnt dictionaries. In: *2014 IEEE International Conference on Image Processing (ICIP)*, pp. 808–812. IEEE (2014)
7. Ahmed, N., Natarajan, T., Rao, K.R.: Discrete cosine transform. *IEEE Trans. Comput.* **23**, 90–93 (1974)
8. Mallat, S.: *A Wavelet Tour of Signal Processing*. Academic Press (1999)
9. Candes, E.J., Donoho, D.L.: Curvelets: A surprisingly effective nonadaptive representation for objects with edges. Technical report, DTIC Document (2000)
10. Do, M.N., Vetterli, M.: The contourlet transform: an efficient directional multiresolution image representation. *IEEE Trans. Image Process.* **14**, 2091–2106 (2005)
11. Labate, D., Lim, W.Q., Kutyniok, G., Weiss, G.: Sparse multidimensional representation using shearlets. In: *Optics & Photonics 2005*, p. 59140U. International Society for Optics and Photonics (2005)
12. Engan, K., Aase, S.O., Husoy, J.H.: Method of optimal directions for frame design. In: *Proceedings of the 1999 IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP 1999*, vol. 5, pp. 2443–2446 (1999)
13. Vidal, R., Ma, Y., Sastry, S.: Generalized principal component analysis (gpca). *IEEE Trans. Pattern Anal. Mach. Intell.* **27**, 1945–1959 (2005)
14. Mairal, J., Bach, F., Ponce, J., Sapiro, G.: Online dictionary learning for sparse coding. In: *Proceedings of the 26th International Conference on Machine Learning*, pp. 1–8 (2009)
15. Tarquino, J., Rueda, A., Romero, E.: A multiscale/sparse representation for diffusion weighted imaging (DWI) super-resolution. In: *2014 IEEE 11th International Symposium on Biomedical Imaging (ISBI)*, pp. 983–986. IEEE (2014)
16. Liu, Y., Liu, S., Wang, Z.: A general framework for image fusion based on multi-scale transform and sparse representation. *Inf. Fusion* **24**, 147–164 (2015)
17. Yin, H.: Sparse representation with learned multiscale dictionary for image fusion. *Neurocomputing* **148**, 600–610 (2015)
18. Mairal, J., Sapiro, G., Elad, M.: Learning multiscale sparse representations for image and video restoration. *Multiscale Model. Simul.* **7**, 214–241 (2008)
19. Mallat, S.G., Zhang, Z.: Matching pursuits with time-frequency dictionaries. *IEEE Trans. Sig. Process.* **41**, 3397–3415 (1993)
20. Pati, Y.C., Rezaiifar, R., Krishnaprasad, P.: Orthogonal matching pursuit: recursive function approximation with applications to wavelet decomposition, pp. 40–44 (1993)
21. Chen, S.S., Donoho, D.L., Saunders, M.A.: Atomic decomposition by basis pursuit. *SIAM Rev.* **43**, 129–159 (2001)
22. Gorodnitsky, I.F., Rao, B.D.: Sparse signal reconstruction from limited data using FOCUSS: a re-weighted minimum norm algorithm. *IEEE Trans. Sig. Process.* **45**, 600–616 (1997)
23. Efron, B., Hastie, T., Johnstone, I., Tibshirani, R., et al.: Least angle regression. *Ann. Stat.* **32**, 407–499 (2004)
24. Le Pennec, E., Mallat, S.: Sparse geometric image representations with bandelets. *IEEE Trans. Image Process.* **14**, 423–438 (2005)

25. Mallat, S.G.: A theory for multiresolution signal decomposition: the wavelet representation. *IEEE Trans. Pattern Anal. Mach. Intell.* **11**, 674–693 (1989)
26. Simoncelli, E.P., Freeman, W.T.: The steerable pyramid: A flexible architecture for multi-scale derivative computation. In: *ICIP*, p. 3444. *IEEE* (1995)
27. Burt, P.J., Adelson, E.H.: The laplacian pyramid as a compact image code. *IEEE Trans. Commun.* **31**, 532–540 (1983)
28. Dong, W., Zhang, L., Lukac, R., Shi, G.: Sparse representation based image interpolation with nonlocal autoregressive modeling. *IEEE Trans. Image Process.* **22**, 1382–1394 (2013)
29. Rubinstein, R., Zibulevsky, M., Elad, M.: Efficient implementation of the K-SVD algorithm using batch orthogonal matching pursuit. *CS Technion* **40**, 1–15 (2008)
30. Tropp, J.A.: Greed is good: algorithmic results for sparse approximation. *IEEE Trans. Inf. Theory* **50**, 2231–2242 (2004)
31. Dabov, K., Foi, A., Katkovnik, V., Egiazarian, K.: Image denoising by sparse 3-d transform-domain collaborative filtering. *IEEE Trans. Image Process.* **16**, 2080–2095 (2007)