

# Trainlets: Dictionary Learning in High Dimensions

Jeremias Sulam, *Student Member, IEEE*,  
Boaz Ophir, Michael Zibulevsky, and Michael Elad, *Fellow, IEEE*

**Abstract**—Sparse representation has shown to be a very powerful model for real world signals, and has enabled the development of applications with notable performance. Combined with the ability to learn a dictionary from signal examples, sparsity-inspired algorithms are often achieving state-of-the-art results in a wide variety of tasks. These methods have traditionally been restricted to small dimensions mainly due to the computational constraints that the dictionary learning problem entails. In the context of image processing, this implies handling small image patches. In this work we show how to efficiently handle bigger dimensions and go beyond the small patches in sparsity-based signal and image processing methods. We build our approach based on a new cropped Wavelet decomposition, which enables a multi-scale analysis with virtually no border effects. We then employ this as the base dictionary within a double sparsity model to enable the training of adaptive dictionaries. To cope with the increase of training data, while at the same time improving the training performance, we present an Online Sparse Dictionary Learning (OSDL) algorithm to train this model effectively, enabling it to handle millions of examples. This work shows that dictionary learning can be up-scaled to tackle a new level of signal dimensions, obtaining large adaptable atoms that we call *Trainlets*.

**Index Terms**—Double-sparsity, K-SVD, Dictionary Learning, Cropped Wavelet, On-Line Learning, Trainlets, Contourlets.

## I. INTRODUCTION

Sparse representations over redundant dictionaries have shown to be a very powerful model for many real world signals, enabling the development of applications with notable performance in many signal and image processing tasks [1]. The basic assumption of this model is that natural signals can be expressed as a sparse linear combination of atoms, chosen from a collection called a *dictionary*. Formally, for a signal  $\mathbf{y} \in \mathbb{R}^n$ , this can be described by  $\mathbf{y} = \mathbf{D}\mathbf{x}$ , where  $\mathbf{D} \in \mathbb{R}^{n \times m}$ , ( $n < m$ ) is a redundant dictionary that contains the atoms as its columns, and  $\mathbf{x} \in \mathbb{R}^m$  is the representation vector.

Given the signal  $\mathbf{y}$ , finding its representation can be done in terms of the following sparse approximation problem:

$$\min_{\mathbf{x}} \|\mathbf{x}\|_0 \quad \text{subject to} \quad \|\mathbf{y} - \mathbf{D}\mathbf{x}\|_2 \leq \epsilon, \quad (1)$$

where  $\epsilon$  is a permitted deviation in the representation accuracy, and the expression  $\|\mathbf{x}\|_0$  is a count of the number of non-zeroes in the vector  $\mathbf{x}$ . The process of solving the above optimization problem is commonly referred to as sparse coding. Solving this problem is in general NP-hard, but several greedy algorithms and other relaxations methods allow us to solve the problem exactly under certain conditions [1] and obtain useful approximate solutions in more general settings. These methods include MP, OMP, BP and FOCUSS among others (see [2] for a review on these topics).

A fundamental element in this problem is the choice of the dictionary  $\mathbf{D}$ . While some analytically-defined dictionaries (or transformations) such as the overcomplete Discrete Cosine Transform (ODCT) or Wavelet dictionaries were used originally, learning the dictionary from signal examples for a specific task has shown to perform significantly better [3]. This adaptivity to the data allows sparsity-inspired algorithms to achieve state-of-the-art results in many tasks. The dictionary learning problem can be written as:

$$\underset{\mathbf{D}, \mathbf{X}}{\operatorname{argmin}} \frac{1}{2} \|\mathbf{Y} - \mathbf{D}\mathbf{X}\|_F^2 \quad \text{subject to} \quad \|\mathbf{x}_i\|_0 \leq p \quad \forall i, \quad (2)$$

where  $\mathbf{Y} \in \mathbb{R}^{n \times N}$  is a matrix containing  $N$  signal examples, and  $\mathbf{X} \in \mathbb{R}^{m \times N}$  are the corresponding sparse vectors, both ordered column wise. Several iterative methods have been proposed to handle this task [4]–[6]. Due to the computational complexity of this problem, all these methods have been restricted to relatively small signals. When dealing with high-dimensional data, the common approach is to partition the signal into small blocks, where dictionary learning is more feasible.

In the context of image processing, small signals imply handling small image patches. Most state-of-the-art methods for image restoration exploit such a localized patch based approach [7]–[9]. In this setting, small overlapping patches ( $7 \times 7$  -  $11 \times 11$ ) are extracted from

the corrupted image and treated relatively independently according to some image model [9], [10], sparse representations being a popular choice [11]–[14]. The full image estimation is then formed by merging together the small restored patches by overlapping and averaging.

Some works have attempted to handle larger two dimensional patches (i.e., greater than  $16 \times 16$ ) with some success. In [15], and later in [16], traditional K-SVD is applied in the Wavelet domain. These works implicitly manage larger patches while keeping the atom dimension small, noting that small patches of Wavelet coefficients translate to large regions in the image domain. In the context of Convolutional Networks, on the other hand, the work in [17] has reported encouraging state-of-art result on patches of size  $17 \times 17$ .

Though adaptable, explicit dictionaries are computationally expensive to apply. Some efforts have been done in designing fast dictionaries that can be both applied and learned efficiently. This requirement implies constraining the degrees of freedom of the explicit matrix in some way, i.e. imposing some structure on the dictionary. One such possibility is the search for adaptable separable dictionaries, as in [18], or the search of a dictionary which is an image in itself as in [19], [20], lowering the degrees of freedom and obtaining (close to) shift invariant atoms. Another, more flexible alternative, has been the pursuit of sparse dictionaries [21], [22]. In these works the dictionary is composed of a multiplication of two matrices, one of which is sparse. The work in [23] takes this idea a step further, composing a dictionary from the multiplication of a sequence of sparse matrices. In the interesting work reported in [24] the dictionary is modeled as a collection of convolutions with sparse kernels, lowering the complexity of the problem and enabling the approximation of popular analytically-defined atoms. All of these works, however, have not addressed dictionary learning on real data of considerably higher dimensions or with a considerably large dataset.

A related but different model from the one posed in Equation (2) is the analysis model [25], [26]. In this framework, a dictionary  $\mathbf{W}$  is learned such that  $\|\mathbf{W}\mathbf{y}\|_0 \ll n$ . A close variant is the Transform Learning model, where it is assumed that  $\mathbf{W}\mathbf{y} \approx \mathbf{x}$  and  $\|\mathbf{x}\|_0 \ll n$ , as presented in [27]. This framework presents interesting advantages due to the very cheap sparse coding stage. An online transform learning approach was presented in [28], and a sparse transform model was presented in [29], enabling the training on bigger image patches. In our work, however, we constrain ourselves to the study of synthesis dictionary models.

We give careful attention to the model proposed in [21]. In this work a double sparse model is proposed by

combining a fixed separable dictionary with an adaptable sparse component. This lowers the degrees of freedom of the problem in Equation (2), and provides a feasible way of treating high dimensional signals. However, the work reported in [21] concentrated on 2D and 3D-DCT as a base-dictionary, thus restricting its applicability to relatively small patches.

In this work we expand on this model, showing how to efficiently handle bigger dimensions and go beyond the small patches in sparsity-based signal and image processing methods. This model provides the flexibility of incorporating multi-scale properties in the learned dictionary, a property we deem vital for representing larger signals. For this purpose, we propose to replace the fixed base dictionary with a new multi-scale one.

We build our approach on *cropped* Wavelets, a multi-scale decomposition which overcomes the limitations of the traditional Wavelet transform to efficiently represent small images (expressed often in the form of severe border effects).

Another aspect that has limited the training of large dictionaries has been the amount of data required and the corresponding amount of computations involved. As the signal size increases, a (significant) increase in the number of training examples is needed in order to effectively learn the inherent data structure. While traditional dictionary learning algorithms require many sweeps of the whole training corpus, this is no longer feasible in our context. Instead, we look to online learning methods, such as Stochastic Gradient Decent (SGD) [30]. These methods have gained prominence in recent years with the advent of big data, and have been used in the context of traditional (unstructured) dictionary learning [6] and in training the special structure of the Image Signature Dictionary [19]. We present an Online Sparse Dictionary Learning (OSDL) algorithm to effectively train the double-sparsity model. This approach allows us to handle very large training sets while using high dimensional signals, achieving faster convergence than the batch alternative and providing a better treatment of local minima, which are abundant in non-convex dictionary learning problems.

To summarize, this paper introduces a novel online dictionary learning algorithm, which builds a structured dictionary based on the double-sparsity format. The base dictionary proposed is a fully-separable cropped Wavelets dictionary that has virtually no boundary effects. The overall dictionary learning algorithm can be trained on a corpus of millions of examples, and is capable of representing images of size  $64 \times 64$  and even more, while keeping the training, the memory, and the computational load reasonable and manageable.

This high-dimensional dictionary learning framework, termed *Trainlets*, shows that global dictionaries for entire images are feasible and trainable. We demonstrate the applicability of the proposed algorithm and its various ingredients in this paper, and we accompany this work with a freely available software package.

This paper is organized as follows. In section II we review sparse dictionary models. In section III we introduce the cropped Wavelets showing their advantages over standard Wavelets, and in section IV we present the Online Sparse Dictionary Learning algorithm. In section V we present several experiments and applications to image processing, comparing the OSDL algorithm to related methods, demonstrating the benefits of our approach. In section VI we conclude the paper.

## II. SPARSE DICTIONARIES

Learning dictionaries for large signals requires adding some constraint to the dictionary, otherwise signal diversity and the number of training examples needed make the problem intractable. Often, these constraints are given in terms of a certain structure. One such approach is the double-sparsity model [21]. In this model the dictionary is assumed to be a multiplication of a fixed operator  $\Phi$  (we will refer to it as the base dictionary) by a sparse adaptable matrix  $\mathbf{A}$ . Every atom in the effective dictionary  $\mathbf{D}$  is therefore a linear combination of few and arbitrary atoms from the base dictionary. Formally, this means that the training procedure requires solving the following problem:

$$\min_{\mathbf{A}, \mathbf{X}} \frac{1}{2} \|\mathbf{Y} - \Phi \mathbf{A} \mathbf{X}\|_F^2 \quad \text{s.t.} \quad \begin{cases} \|\mathbf{x}_i\|_0 \leq p & \forall i \\ \|\mathbf{a}_j\|_0 = k & \forall j \end{cases}. \quad (3)$$

Note that the number of columns in  $\Phi$  and  $\mathbf{A}$  might differ, allowing flexibility in the redundancy of the effective dictionary. The authors in [21] used an over-complete Discrete Cosine Transform (ODCT) as the base dictionary in their experiments. Using Wavelets was proposed but never implemented due both to implementation issues (the traditional Wavelet transform is not entirely separable) and to the significant border-effects Wavelets have in small-to-medium sized patches. We address both of these issues in the following section.

As for the training of such a model, the update of the dictionary is now constrained by the number of non-zeros in the columns of  $\mathbf{A}$ . In [21] a variant of the K-SVD algorithm (termed Sparse K-SVD) was proposed for updating the dictionary. As the work in [4], this is a batch method that updates every atom sequentially. In the context of the double-sparsity structure, this task is converted into a sparse-coding problem, and approximated by the greedy OMP algorithm.

In the recent inspiring work reported in [23] the authors extended the double-sparsity model to a scenario where the base dictionary itself is a multiplication of several sparse matrices, that are to be learned. While this structure allows for a clear decrease in the computational cost of applying the dictionary, its capacity to treat medium-size problems is not explored. The proposed algorithm involves a hierarchy of matrix factorizations with multiple parameters to be set, such as the number of levels and the sparsity of each level.

## III. A NEW WAVELETS DICTIONARY

The double sparsity model relies on a base-dictionary which should be computationally efficient to apply. The ODCT dictionary has been used for this purpose in [21], but its applicability to larger signal sizes is weak. Indeed, as the patch size grows – getting closer to an image size – the more desirable a multi-scale analysis framework becomes. The separability of the base dictionary provides a further decrease in the computational complexity. Applying two (or more) 1D dictionaries on each dimension separately is typically much more efficient than an equivalent non-separable multi-dimensional dictionary. We will combine these two characteristics as guidelines in the design of the base dictionary for our model.

### A. Optimal Extensions and Cropped Wavelets

The two dimensional Wavelet transform has shown to be very effective in sparsifying natural (normal sized) images. When used to analyze small or medium sized images, not only is the number of possible decomposition scales limited, but more importantly the border effects become a serious limitation. Other works have pointed out the importance of the boundary conditions in the context of deconvolution [31], [32]. However, our approach is different from these, as we will focus on the basis elements rather than on the signal boundaries, and in the pursuit of the corresponding coefficients.

In order to build (bi-)orthogonal Wavelets over a finite (and small) interval, one usually assumes their periodic or symmetric extension onto an infinite axis. A third alternative, zero-padding, assumes the signal is zero outside of the interval. However, none of these alternatives provides an optimal approximation of the signal borders. In general, all these methods do not preserve their vanishing moments at the boundary of the interval, leading to additional non-zero coefficients corresponding to the basis functions that overlap with the boundaries [33]. An alternative is to modify the Wavelet filters such that they preserve their vanishing moments at the borders of the interval, although constructing

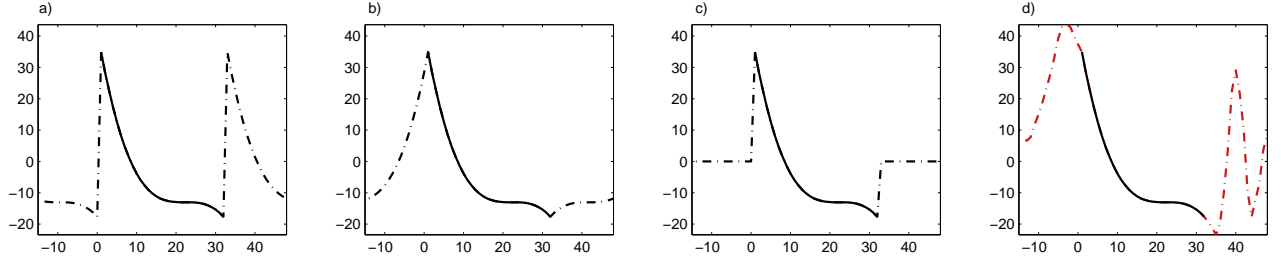


Fig. 1. Different border treatments: a) periodic, b) symmetric, c) zero-padding, and d) the resulting optimized extension signal  $\hat{\mathbf{f}} = \mathbf{W}_s \mathbf{g}_w$ .

such Wavelets while preserving their orthogonality is complicated [34].

We begin our derivation by looking closely at the zero-padding case. Let  $\mathbf{f} \in \mathbf{R}^n$  be a finite signal. Consider  $\bar{\mathbf{f}} = \mathbf{P}\mathbf{f}$ , the zero-padded version of  $\mathbf{f}$ , where  $\mathbf{P} \in \mathbf{R}^{L \times n}$ ,  $L > n$  ( $L$  is “big enough”). Considering the Wavelet analysis matrix  $\mathbf{W}_a$  of size  $L \times L$ , the Wavelet representation coefficients are obtained by applying the Discrete Wavelet Transform (DWT) to  $\bar{\mathbf{f}}$ , which can be written as  $\mathbf{g}_w = \mathbf{W}_a \bar{\mathbf{f}}$ . Note that this is just a projection of the (zero-padded) signal onto the orthogonal Wavelet atoms. As for the inverse transform, the padded signal is recovered by applying the inverse Wavelet transform or Wavelet synthesis operator  $\mathbf{W}_s$  ( $\mathbf{W}_s = \mathbf{W}_a^T$ , assuming orthogonal Wavelets), of size  $L \times L$  to the coefficients  $\mathbf{g}_w$ . Lastly, the padding is discarded (multiplying by  $\mathbf{P}^T$ ) to obtain the final signal in the original finite interval:

$$\hat{\mathbf{f}} = \mathbf{P}^T \mathbf{W}_s \mathbf{g}_w = \mathbf{P}^T \mathbf{W}_s (\mathbf{W}_a \mathbf{P} \mathbf{f}) = \mathbf{f}. \quad (4)$$

Zero-padding is not an option of preference because it introduces discontinuities in the function  $\bar{\mathbf{f}}$  that result in large (and many) Wavelet coefficients, even if  $\mathbf{f}$  is smooth inside the finite interval. This phenomenon can be understood from the following perspective: we are seeking the representation vector  $\mathbf{g}_w$  that will satisfy the perfect reconstruction of  $\mathbf{f}$ ,

$$\mathbf{P}^T \mathbf{W}_s \mathbf{g}_w = \mathbf{f}. \quad (5)$$

The matrix  $\mathbf{P}^T \mathbf{W}_s$  serves here as the effective dictionary that multiplies the representation in order to recover the signal. This relation is an under-determined linear system of equations with  $n$  equations and  $L$  unknowns, and thus it has infinitely many possible solutions.

In fact, zero padding chooses a very specific solution to the above system, namely,  $\mathbf{g}_w = \mathbf{W}_a \mathbf{P} \mathbf{f}$ . This is nothing but the projection of the signal onto the adjoint of the above-mentioned dictionary, since  $\mathbf{W}_a \mathbf{P} = (\mathbf{P}^T \mathbf{W}_s)^T$ . While this is indeed a feasible solution, such a solution is expected to have many non-zeros if the atoms are strongly correlated. This indeed occurs for the finite-support Wavelet atoms that intersect the borders, and which are cropped by  $\mathbf{P}^T$ .

To overcome this problem, we propose the following alternative optimization objective:

$$\mathbf{g}_w = \arg \min_{\mathbf{g}} \|\mathbf{g}\|_0 \quad \text{s.t.} \quad \mathbf{P}^T \mathbf{W}_s \mathbf{g} = \mathbf{f}, \quad (6)$$

i.e., seeking the sparsest solution to this under-determined linear system. Note that in performing this pursuit, we are implicitly extending the signal  $\mathbf{f}$  to become  $\bar{\mathbf{f}} = \mathbf{W}_s \mathbf{g}_w$ , which is the smoothest possible with respect to the Wavelet atoms (i.e., it is sparse under the Wavelet transform). At the same time, we keep using the original Wavelet atoms with all their properties, including their vanishing moments. On the other hand, we pay the price of performing a pursuit instead of a simple back-projection. In particular, we use OMP to approximate the solution to this sparse coding problem. To conclude, our treatment of the boundary issue is obtained by applying the cropped Wavelets dictionary  $\mathbf{W}_c = \mathbf{P}^T \mathbf{W}_s$ , and seeking the sparsest representation with respect to it, implicitly obtaining an extension of  $\mathbf{f}$  without boundary problems.

To illustrate our approach, in Fig. 1 we show the typical periodic, symmetric and zero-padding border extensions applied to a random smooth function, as well as the ones obtained by our method. As can be seen, this extension – which is nothing else than Wavelet atoms that fit in the borders in a natural way – guarantees not to create discontinuities which result in denser representations<sup>1</sup>. Note that we will not be interested in the actual extensions explicitly in our work.

To provide further evidence on the better treatment of the borders by the cropped Wavelets, we present the following experiment. We construct 1,000 random smooth functions  $\mathbf{f}$  of length 64 (3rd degree polynomials), and introduce a random step discontinuity at sample 32. These signals are then normalized to have unit  $l_2$ -norm. We approximate these functions with only 5 Wavelet coefficients, and measure the energy of the point-wise (per sample) error (in  $l_2$ -sense) of the reconstruction.

<sup>1</sup>A similar approach was presented in [35] in the context of compression. The authors proposed to optimally extend the borders of an irregular shape in the sense of minimal  $l_1$ -norm of the representation coefficients under a DCT transform.



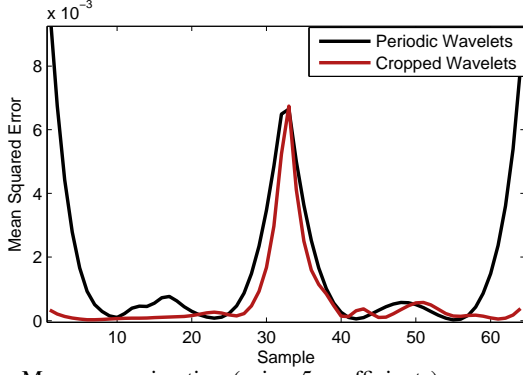


Fig. 2. Mean approximation (using 5 coefficients) error *per sample* of smooth functions of length 64 with a discontinuity at sample 32.

Fig. 2 shows the mean distribution of these errors<sup>2</sup>. As expected, the discontinuity at the center introduces a considerable error. However, the traditional (periodic) Wavelets also exhibit substantial errors at the borders. The proposed cropped Wavelets, on the other hand, manage to reduce these errors by avoiding the creation of extra discontinuities.

Practically speaking, the proposed cropped Wavelet dictionary can be constructed by taking a Wavelet synthesis matrix for signals of length  $L$  and cropping it. Also, and because we will be making use of greedy pursuit methods, each atom is normalized to have unit  $l_2$  norm. This way, the cropped Wavelets dictionary can be expressed as

$$\Phi_1^c = \mathbf{P}^T \mathbf{W}_s \mathbf{W},$$

where  $\mathbf{W}$  is a diagonal matrix of size  $L \times L$  with values such that each atom (column) in  $\Phi_1^c$  (of size  $n \times L$ ) has a unit norm<sup>3</sup>. The resulting transform is no longer orthogonal, but this – now redundant – Wavelet dictionary solves the borders issues of traditional Wavelets enabling for a lower approximation error.

Just as in the case of zero-padding, the redundancy obtained depends on the dimension of the signal, the number of decomposition scales and the length of the support of the Wavelet filters (refer to [33] for a thorough discussion). In practice, we set  $L = 2^{\lceil \log_2(n) \rceil + 1}$ ; i.e., twice the closest higher power of 2 (which reduces to  $L = 2n$  if  $n$  is a power of two, yielding a redundancy of at most 2) guaranteeing a sufficient extension of the borders.

<sup>2</sup>The  $m$ -term approximation with Wavelets is performed with the traditional non-linear approximation scheme. In this framework, orthogonal Wavelets with periodic extensions perform better than symmetric extensions or zero-padding, which we therefore omit from the comparison. We used for this experiment Daubechies Wavelets with 13 taps. All random variables were chosen from Gaussian distributions.

<sup>3</sup>Because the atoms in  $\mathbf{W}_s$  are compactly supported, some of them may be identically zero in the central  $n$  samples. These are discarded in the construction of  $\Phi_1^c$ .

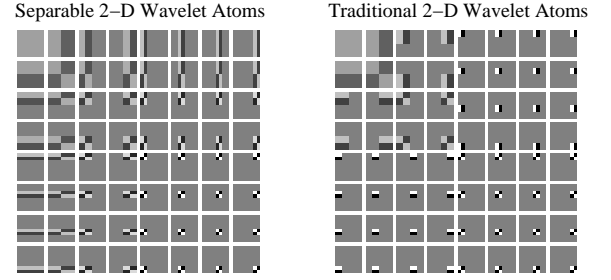


Fig. 3. 2-D atoms of the Wavelet (Haar) transform for patches of size  $8 \times 8$  – the separable versus the traditional construction.

### B. A Separable 2-D Extension

The one-dimensional Wavelet transform is traditionally extended to treat two-dimensional signals by constructing two-dimensional atoms as the separable product of two one-dimensional ones, per scale [33]. This yields three two-dimensional Wavelet functions at each scale  $j$ , implying a decomposition which is only separable per scale. This means cascading this two-dimensional transform on the approximation band at every scale.

An alternative extension is a completely separable construction. Considering all the basis elements of the 1-D DWT (in *all* scales) arranged column-wise in the matrix  $\Phi_1$ , the 2-D separable transform can be represented as the Kronecker product  $\Phi_2 = \Phi_1 \otimes \Phi_1$ . This way, all properties of the transform  $\Phi_1$  translate to each of the dimensions of the 2-dimensional signal on which  $\Phi_2$  is applied. Now, instead of cascading down a two-dimensional decomposition, the same 1-D Wavelet transform is applied first to all the columns of the image and then to all the rows of the result (or vice versa). In relatively small images, this alternative is simpler and faster to apply compared to the traditional cascade. This modification is not only applicable to the traditional Wavelet transform, but also to the cropped Wavelets dictionary introduced above. In this 2-D set-up, both vertical and horizontal borders are implicitly extended to provide a sparser Wavelet representation.

We present in Fig. 3 the 2-D atoms of the Wavelet (Haar) Transform for signals of size  $8 \times 8$  as an illustrative example. The atoms corresponding to the coarsest decomposition scale and the diagonal bands are the same in both separable and non-separable constructions. The difference appears in the vertical and horizontal bands (at the second scale and below). In the separable case we see elongated atoms, mixing a low scale in one direction with high scale in the other.

### C. Approximation of Real World Signals

While it is hard to rank the performance of separable versus non-separable *analytical* dictionaries or transforms in the general case, we have observed that

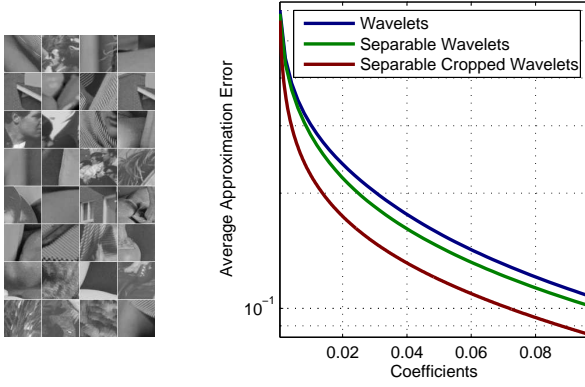


Fig. 4. Left: Random set of some of the images used for the M-Term approximation Experiment. Right: M-Term approximation by the traditional 2-D Wavelets and the separable and cropped Wavelets on real images of size  $64 \times 64$ .

the separable Wavelet transform provides sparser representations than the traditional 2-D decomposition on small-medium size images. To demonstrate this, we take 1,000 image patches of size  $64 \times 64$  from popular test images, and compare the m-term approximation achieved by the regular two-dimensional Wavelet transform, the completely separable Wavelet transform and our separable and cropped Wavelets. A small subset of these patches is presented on the left of Fig. 4. These large patches are in themselves small images, exhibiting the complex structures characteristic of real world images.

As we see from the results in Fig. 4 (right), the separability provides some advantage over regular Wavelets in representing the image patches. Furthermore, the proposed separable cropped Wavelets give an even better approximation of the data with fewer coefficients.

Before concluding this section, we make the following remark. It is well known that Wavelets (separable or not) are far from providing an optimal representation for general images [33], [36], [37]. Nonetheless, in this work these basis functions will be used only as the base dictionary, while our learned dictionary will consist of linear combinations thereof. It is up to the learning process to close the gap between the sub-optimal representation capability of the Wavelets, and the need for a better two dimensional representation that takes into account edge orientation, scale invariance, and more.

#### IV. ONLINE SPARSE DICTIONARY LEARNING

As seen previously, the *de-facto* method for training the doubly sparse model has been a batch-like process. When working with higher dimensional data, however, the required amount of training examples and the corresponding computational load increase. In this *big-data* (or *medium-data*) scenario, it is often unfeasible or undesired to perform several sweeps over the entire data set. In some cases, the dimensionality and amount

of data might restrict the learning process to only a couple of iterations. In this regime of work it may be impossible to even store all training samples in memory during the training process. In an extreme online learning set-up, each data sample is seen only once as new data flows in. These reasons lead naturally to the formulation of an online training method for the double-sparsity model. In this section, we first introduce a dictionary learning method based on the Normalized Iterative Hard-Thresholding algorithm [38]. We then use these ideas to propose an Online Sparse Dictionary Learning (OSDL) algorithm based on the popular Stochastic Gradient Descent technique, and show how it can be applied efficiently to our specific dictionary learning problem.

##### A. NIHT-based Dictionary Learning

A popular practice in dictionary learning, which has been shown to be quite effective, is to employ a block coordinate minimization over this non-convex problem. This often reduces to alternating between a sparse coding stage, throughout which the dictionary is held constant, and a dictionary update stage in which the sparse coefficients (or their support) are kept fixed. We shall focus on the second stage, as the first remains unchanged, essentially applying sparse coding to a group of examples. Embarking from the objective as given in Equation (3), the problem to consider in the dictionary update stage is the following:

$$\min_{\mathbf{A}} \underbrace{\frac{1}{2} \|\mathbf{Y} - \Phi \mathbf{A} \mathbf{X}\|_F^2}_{f(\mathbf{A})} \quad \text{s.t.} \quad \|\mathbf{a}_j\|_0 = k \quad \forall j, \quad (7)$$

where  $\Phi$  is the base dictionary of size  $n \times L$  and  $\mathbf{A}$  is a matrix of size  $L \times m$  with  $k$  non-zeros per column. Many dictionary learning methods undertake a sequential update of the atoms in the dictionary ([4], [6], [21]). Following this approach, we can consider  $m$  minimization problems of the following form:

$$\min_{\mathbf{a}_j} \underbrace{\frac{1}{2} \|\mathbf{E}_j - \Phi \mathbf{a}_j \mathbf{x}_j^T\|_F^2}_{f(\mathbf{a}_j)} \quad \text{s.t.} \quad \|\mathbf{a}_j\|_0 = k, \quad (8)$$

where  $\mathbf{E}_j$  is the error given by  $\mathbf{Y} - \sum_{i \neq j} \Phi \mathbf{a}_i \mathbf{x}_i^T$  and  $\mathbf{x}_i^T$  denotes the  $i$ -th row of  $\mathbf{X}$ . This problem produces the  $j$ -th column in  $\mathbf{A}$ , and thus we sweep through  $j = 1, \dots, m$  to update all of  $\mathbf{A}$ .

The Normalized Iterative Hard-Thresholding (NIHT) [38] algorithm is a popular sparse coding method in the context of Compressed Sensing [39]. This method can be understood as a projected gradient descent algorithm. We can propose a dictionary update based on the same concept. Note that we could rewrite the cost function

in Equation (8) as  $f(\mathbf{a}_j) = \frac{1}{2} \|\mathbf{E}_j - \mathcal{H}_j \mathbf{a}_j\|_F^2$ , for an appropriate operator  $\mathcal{H}_j$ . Written in this way, we can perform the dictionary update in terms of the NIHT by iterating:

$$\mathbf{a}_j^{t+1} = \mathcal{P}_k [\mathbf{a}_j^t - \eta_j^t \mathcal{H}_j^* (\mathbf{E}_j - \mathcal{H}_j \mathbf{a}_j^t)], \quad (9)$$

where  $\mathcal{H}_j^*$  is the adjoint of  $\mathcal{H}_j$ ,  $\mathcal{P}_k$  is a Hard-Thresholding operator that keeps the  $k$  largest non-zeros (in absolute value), and  $\eta_j^t$  is an appropriate step-size. This algorithm iterates over Equation (9) until convergence *per atom* in the dictionary update stage.

The choice of the step size is critical. Noting that  $\mathcal{H}_j^* (\mathbf{Y} - \mathcal{H}_j \mathbf{a}_j) = \nabla f(\mathbf{a}_j)$ , in [38] the authors propose to set this parameter per iteration as:

$$\eta_j^t = \frac{\|\nabla f(\mathbf{a}_j^t)_{S_j}\|_F^2}{\|\mathcal{H} \nabla f(\mathbf{a}_j^t)_{S_j}\|_F^2}, \quad (10)$$

where  $S_j$  denotes the support of  $\mathbf{a}_j^t$ . With this step size, the estimate  $\hat{\mathbf{a}}^{t+1}$  is obtained by performing a gradient step and hard-thresholding as in Equation (10). Note that if the support of  $\hat{\mathbf{a}}_j^{t+1}$  and  $\mathbf{a}_j^t$  are the same, setting  $\eta_j^t$  as in Equation (10) is indeed optimal, as it is the minimizer of the quadratic cost w.r.t.  $\eta_j^t$ . In this case, we simply set  $\mathbf{a}_j^{t+1} = \hat{\mathbf{a}}_j^{t+1}$ . If the support changes after applying  $\mathcal{P}_k$ , however, the step-size must be diminished until a condition is met, guaranteeing a decrease in the cost function<sup>4</sup>. Following this procedure, the work reported in [38] shows that the algorithm in Equation (9) is guaranteed to converge to a local minimum of the problem in Equation (8).

Consider now the algorithm given by iterating between 1) sparse coding of all examples in  $\mathbf{Y}$ , and 2) atom-wise dictionary update with NIHT in Equation (8). An important question that arises is: will this simple algorithm converge? Let us assume that the pursuit succeeds, obtaining the sparsest solution for a given sparse dictionary  $\mathbf{A}$ , which can indeed be guaranteed under certain conditions. Moreover, pursuit methods like OMP, Basis Pursuit and FOCUSS perform very well in practice when  $k \ll n$  (refer to [2] for a thorough review). For the cases where the theoretical guarantees are not met, we can adopt an external interference approach by comparing the best solution using the support obtained in the previous iteration to the one proposed by the new iteration of the algorithm, and choosing the best one. This small modification guarantees a decrease in the cost function at every sparse coding step. The atom-wise update of the dictionary is also guaranteed to converge to a local minimum for the above mentioned choice

<sup>4</sup>The step size is decreased by  $\eta_j^t = c \eta_j^t$ , where  $c < 1$ . We refer the reader to [39] and [38] for further details.

---

### Algorithm 1: Stochastic NIHT for Sparse Dictionary Learning.

---

**Data:** Training samples  $\{\mathbf{y}_i\}$ , base-dictionary  $\Phi$ , initial sparse matrix  $\mathbf{A}^0$

```

for  $i = 1, \dots, Iter$  do
    Draw  $\mathbf{y}_i$  at random;
     $\mathbf{x}_i \leftarrow$  Sparse Code ( $\mathbf{y}_i, \Phi, \mathbf{A}^i$ );
     $\mathcal{S}_i = \text{Support}(\mathbf{x}_i)$ ;
    for  $j = 1, \dots, |\mathcal{S}_i|$  do
        Update  $\mathbf{a}_{\mathcal{S}(j)}^{i+1}$  with Equation (11) and step
        size  $\frac{\eta_{\mathcal{S}(j)}^*}{1+i/T}$ ;
    end
end

```

**Result:** Sparse Dictionary  $\mathbf{A}$

---

of step sizes. Performing a series of these alternating minimization steps ensures a monotonic reduction in the original cost function in Equation (2), which is also bounded from below, and thus convergence to a fixed point is guaranteed.

Before moving on, a word on the characteristics of trained dictionaries is in place. The recovery guarantees of pursuit methods is generally formulated in terms of properties of the dictionary, such as its mutual coherence or its Restricted Isometry Property (RIP) [1]. While dictionary learning provides better and sparser representations for real data, this adaptive process generally deteriorates these measures. A trained dictionary does not typically exhibit low correlation between its atoms, and so the corresponding results (which are worst-case scenario analyses) say very little about the quality of the obtained dictionary. As we shall see in the results section, this does not imply a deterioration of its practical performance; on the contrary, their effectiveness in image representation and restoration is greatly improved.

### B. From Batch to Online Learning

As noted in [6], [19], it is not compulsory to accumulate all the examples to perform an update in the gradient direction. Instead, we turn to a stochastic (projected) gradient descent approach. In this scheme, instead of computing the expected value of the gradient by the sample mean over all examples, we estimate this gradient over a single randomly chosen example  $\mathbf{y}_i$ . We then update the atoms of the dictionary based on this

estimation using:

$$\mathbf{a}_j^{t+1} = \mathcal{P}_k [\mathbf{a}_j^t - \eta^t \nabla f(\mathbf{a}_j^t, \mathbf{y}_i, \mathbf{x}_i,)] . \quad (11)$$

Since these updates might be computationally costly (and because we are only performing an alternating minimization over problem (3)), we might stop after a few iterations of applying Equation (11). We also restrict this update to those atoms that are used by the current example (since others have no contribution in the corresponding gradient). In addition, instead of employing the step size suggested by the NIHT algorithm, we employ the common approach of using decreasing step sizes throughout the iterations, which has been shown beneficial in stochastic optimization [30]. To this end, and denoting by  $\eta_j^*$  the step size resulting from the NIHT, we employ an effective learning rate of  $\frac{\eta_j^*}{1+t/T}$ , with a manually set parameter  $T$ . This modification does not compromise the guarantees of a decrease in the cost function (for the given random sample  $i$ ), since this factor is always smaller than one. We outline the basic stages of this method in Algorithm 1.

An important question that now arises is whether shifting from a batch training approach to this online algorithm preserves the convergence guarantees described above. Though plenty is known in the field of stochastic approximations, most of the existing results address convergence guarantees for convex functions, and little is known in this area regarding projected gradient algorithms [40]. For non-convex cases, convergence guarantees still demand the cost function to be differentiable with continuous derivatives [30]. In our case, the  $l_0$  pseudo-norm makes a proof of convergence challenging, since the problem becomes not only non-convex but also (highly) discontinuous.

That said, one could re-formulate the dictionary learning problem using a non-convex but continuous and differentiable penalty function, moving from a constrained optimization problem to an unconstrained one. We conjecture that convergence to a fixed point of this problem can be reached under the mild conditions described in [30]. Despite these theoretical benefits, we choose to maintain our initial formulation in terms of the  $l_0$  measure for the sake of simplicity (note that we need no parameters other than the target sparsity). Practically, we saw in all our experiments that convergence is reached, providing numerical evidence for the behavior of our algorithm.

### C. OSDL In Practice

We now turn to describe a variant of the method described in Algorithm 1, and outline other implementation details. The atom-wise update of the dictionary,

---

### Algorithm 2: Online Sparse Dictionary Learning (OSDL) algorithm.

---

**Data:** Training samples  $\{\mathbf{y}_i\}$ , base-dictionary  $\Phi$ , initial sparse matrix  $\mathbf{A}^0$

Initialization:  $\mathbf{G}_\Phi = \Phi^T \Phi$ ;  $\mathbf{U} = \mathbf{0}$  ;

**for**  $t = 1, \dots, T$  **do**

Draw a mini-batch  $\mathbf{Y}_t$  at random;  
 $\mathbf{X}_t \leftarrow \text{Sparse Code}(\mathbf{Y}_t, \Phi, \mathbf{A}^t, \mathbf{G}^t)$ ;  
 $\eta^t = \|\nabla f(\mathbf{A}_S^t)\|_F / \|\Phi \nabla f(\mathbf{A}_S^t) \mathbf{X}_t^S\|_F$ ;  
 $\mathbf{U}_S^{t+1} = \gamma \mathbf{U}_S^t + \eta^t \nabla f(\mathbf{A}_S^t)$ ;  
 $\mathbf{A}_S^{t+1} = \mathcal{P}_k [\mathbf{A}_S^t - \mathbf{U}_S^{t+1}]$ ;  
 Update columns and rows of  $\mathbf{G}$  by  
 $(\mathbf{A}^{t+1})^T \mathbf{G}_\Phi \mathbf{A}_S^{t+1}$

**end**

**Result:** Sparse Dictionary  $\mathbf{A}$

---

while providing a specific step-size, is computationally slower than a global update. In addition, guaranteeing a decreasing step in the cost function implies a line-search per atom that is costly. For this reason we propose to replace this stage by a global dictionary update of the form

$$\mathbf{A}^{t+1} = \mathcal{P}_k [\mathbf{A}^t - \eta^t \nabla f(\mathbf{A}^t)] , \quad (12)$$

where the thresholding operator now operates in each column of its argument. While we could maintain a NIHT approach in the choice of the step-size in this case as well, we choose to employ

$$\eta^* = \frac{\|\nabla f(\mathbf{A}_S)\|_F}{\|\Phi \nabla f(\mathbf{A}_S) \mathbf{X}\|_F} . \quad (13)$$

Note that this is the square-root of the value in Equation (10) and it may appear as counter-intuitive. We shall present a numerical justification of this choice in the following section.

Secondly, instead of considering a single sample  $\mathbf{y}_t$  per iteration, a common practice in stochastic gradient descent algorithms is to consider mini-batches  $\{\mathbf{y}_i\}$  of  $N$  examples arranged in the matrix  $\mathbf{Y}_t$ . As explained in detail in [41], the computational cost of the OMP algorithm can be reduced by precomputing (and storing) the Gram matrix of the dictionary  $\mathbf{D}$ , given by  $\mathbf{G} = \mathbf{D}^T \mathbf{D}$ . In a regular online learning scheme, this would be infeasible due to the need to recompute this matrix for each example. In our case, however, the matrix needs only to be updated once per mini-batch. Furthermore, only a few atoms get updated each time. We exploit this



by updating only the respective rows and columns of the matrix  $\mathbf{G}$ . Moreover, this update can be done efficiently due to the sparsity of the dictionary  $\mathbf{A}$ .

Stochastic algorithms often introduce different strategies to regularize the learning process and try to avoid local minimum traps. In our case, we incorporate in our algorithm a momentum term  $\mathbf{U}^t$  controlled by a parameter  $\gamma \in [0, 1]$ . This term helps to attenuate oscillations and can speed up the convergence by incorporating information from the previous gradients. This algorithm, termed Online Sparse Dictionary Learning (OSDL) is depicted in Algorithm 2. In addition, many dictionary learning algorithms [4], [6] include the replacement of (almost) unused atoms and the pruning of similar atoms. We incorporate these strategies here as well, checking for such cases once every few iterations.

#### D. Complexity Analysis

We now turn to address the computational cost of the proposed online learning scheme<sup>5</sup>. As was thoroughly discussed in [21], the sparse dictionary enables an efficient sparse coding step. In particular, any multiplication by  $\mathbf{D}$ , or its transpose, has a complexity of  $\mathcal{T}_D = \mathcal{O}(km + \mathcal{T}_\Phi)$ , where  $m$  is the number of atoms in  $\Phi$  (assume for simplicity  $\mathbf{A}$  square),  $k$  is the atom sparsity and  $\mathcal{T}_\Phi$  is the complexity of applying the base dictionary. For the separable case, this reduces to  $\mathcal{T}_\Phi = \mathcal{O}(n\sqrt{m})$ .

Using a sparse dictionary, the sparse coding stage with OMP (in its Cholesky implementation) is  $\mathcal{O}(pn\sqrt{m} + pkm)$  per example. Considering  $N$  examples in a mini-batch, and assuming  $n \propto m$  and  $p \propto n$ , we obtain a complexity of  $\mathcal{O}(Nn^2(\sqrt{m} + p))$ .

Moving to the update stage in the OSDL algorithm, calculating the gradient  $\nabla f(A_S)$  has a complexity of  $\mathcal{T}_{\nabla f} = \mathcal{O}((k|\mathcal{S}| + n\sqrt{m})N)$ , and so does the calculation of the step size. Recall that  $\mathcal{S}$  is the set of atoms used by the current samples, and that  $|\mathcal{S}| < m$ ; i.e., the update is applied only on a subset of all the atoms. Updating the momentum variable grows as  $\mathcal{O}(|\mathcal{S}|m)$ , and the hard thresholding operator is  $\mathcal{O}(|\mathcal{S}|m \log(m))$ . In a pessimistic approach, assume  $|\mathcal{S}| \propto n$ .

Putting these elements together, the OSDL algorithm has a complexity of  $\mathcal{O}(Nn^2(\sqrt{m} + k) + m^2 \log(m))$  per mini-batch. The first term depends on the number of examples per mini-batch, and the second one depends only on the size of the dictionary. For high dimensions (large  $n$ ), the first term is the leading one. Clearly, the number of non-zeros per atom  $k$  determines the

computational complexity of our algorithm. While in this study we do not address the optimal way of scaling  $k$ , experiments shown hereafter suggest that its dependency with  $n$  might in fact be less than linear. The sparse dictionary provides a computational advantage over the online learning methods using explicit dictionaries, such as [6], which have complexity of  $\mathcal{O}(Nn^3)$ .

## V. EXPERIMENTS

In this section we present a number of experiments to illustrate the behaviour of the method presented in the previous section. We start with a detailed experiment on learning an image-specific dictionary. We then move on to demonstrations on image denoising and image compression. Finally we tackle the training of universal dictionaries on millions of examples in high dimensions.

#### A. Image-Specific Dictionary Learning

To test the behaviour of the proposed approach, we present the following experiment. We train an adaptive sparse dictionary in three setups of increasing dimension: with patches of size  $12 \times 12$ ,  $20 \times 20$  and  $32 \times 32$ , all extracted from the popular image Lena, using a fixed number of non-zeros in the sparse coding stage (4, 10 and 20 non-zeros, respectively). We also repeat this experiment for different levels of sparsity of the dictionary  $\mathbf{A}$ . We employ the OSDL algorithm, as well as the method presented in Algorithm 1 (in its mini-batch version, for comparison). We also include the results by Sparse K-SVD, which is the classical (batch) method for the double sparsity model, and the popular Online Dictionary Learning (ODL) algorithm [42]. Note that this last method is an online method that trains a dense (full) dictionary. Training is done on 200,000 examples, leaving 30,000 as a test set.

The sparse dictionaries use the cropped Wavelets as their operator  $\Phi$ , built using the Symlet Wavelet with 8-taps. The redundancy of this base dictionary is 1.75 (in 1-D), and the matrix  $\mathbf{A}$  is set to be square, resulting in a total redundancy of just over 3. For a fair comparison, we initialize the ODL method with the same cropped Wavelets dictionary. All methods use OMP in the sparse coding stage. Also, note that the ODL<sup>6</sup> algorithm is implemented entirely in C, while in our case this is only true for the sparse coding, giving the ODL somewhat of an advantage in run-time.

The results are presented in Fig. 5, showing the representation error on the test set, where each marker corresponds to an epoch. The atom sparsity refers to the

<sup>5</sup>We analyze the complexity of just the OSDL for simplicity. The analysis of Algorithm 1 is similar, adding the complexity of the line search of the step sizes.

<sup>6</sup>We used the publicly available SPArse Modeling Software package, at <http://spams-devel.gforge.inria.fr/>.

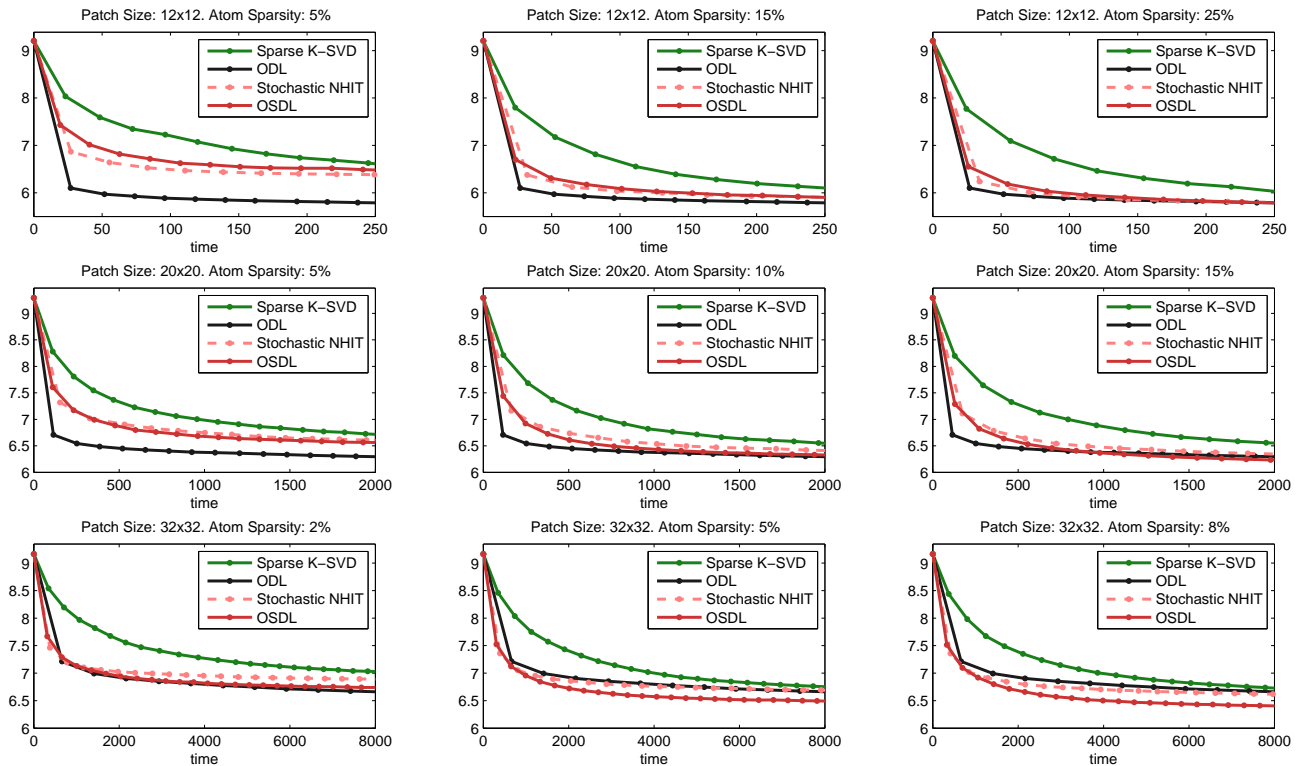


Fig. 5. Experiment 1: Dictionary learning by Sparse K-SVD, by the Stochastic NIHT presented in Algorithm 1, the ODL algorithm [42] and by the Online Sparse Dictionary Learning (OSDL).

number of non-zeros per column of  $\mathbf{A}$  with respect to the signal dimension (i.e., 5% in the  $12 \times 12$  case implies 7 non-zeros). Several conclusions can be drawn from these results. First, as expected, the online approaches provide a much faster convergence than the batch alternative. For the low dimensional case, there is little difference between Algorithm 1 and the OSDL, though this difference becomes more prominent as the dimension increases. In these cases, not only does Algorithm 1 converge slower but it also seems to be more prone to local minima.

As the number of non-zeros per atom grows, the representation power of our sparse dictionary increases. In particular, OSDL achieves the same performance as ODL for an atom sparsity of 25% for a signal dimension of 144. Interestingly, OSDL and ODL achieve the same performance for decreasing number of non-zeros in  $\mathbf{A}$  as the dimension increases: 10% for the  $20 \times 20$  case and  $\approx 2\%$  for the  $32 \times 32$ . In this higher dimensional setting, not only does the sparse dictionary provide faster convergence but it also achieves a lower minimum. The lower degrees of freedom of the sparse dictionary prove beneficial in this context<sup>7</sup>, where the amount of training data is limited and perhaps insufficient to train a full dictionary. This example suggests that indeed  $k$  could

grow slower than linearly with the dimension  $n$ .

Before moving on, we want to provide some empirical evidence to support the choice of the step size in the OSDL algorithm. In Fig. 6 we plot the atom-wise step sizes obtained by Algorithm 1,  $\eta_j^*$  (i.e., the optimal values from the NIHT perspective), together with their mean value, as a function of the iterations for the  $12 \times 12$  case for illustration. In addition, we show the global step sizes of OSDL as in Equation (13). As can be seen, this choice provides a fair approximation to the mean of the individual step sizes. Clearly, the square of this value would be too conservative, yielding very small step sizes and providing substantially slower convergence.

### B. Image Restoration Demonstration

In the context of image restoration, most state-of-the-art algorithms take a patch-based approach. While the different algorithms differ in the models they enforce on the corrupted patches (or the prior they chose to consider, in the context a Bayesian formulation) the general scheme remains very much the same: overlapping patches are extracted from the degraded image, then restored more or less independently, before being merged back together by averaging. Though this provides an effective option, this *locally-focused* approach is far from being optimal. As noted in several recent works ([16], [43], [44]), not looking at the image as a whole

<sup>7</sup>Note that this limitation needed to be imposed for a comparison with Sparse K-SVD. Further along this section we will present a comparison without this limitation.

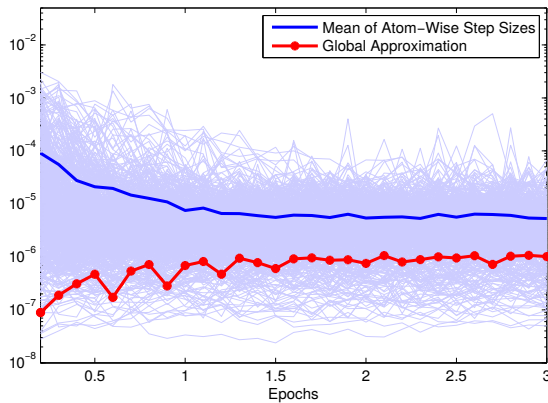


Fig. 6. Step sizes  $\eta_j^*$  obtained by the atom-wise NIHT algorithm together with their mean value, and the global approximation by OSDL.

causes inconsistencies between adjacent patches which often result in texture-like artifacts. A possible direction to seek for a more global outlook is, therefore, to allow for bigger patches.

We do not intend to provide a complete image restoration algorithm in this paper. Instead, we will show that benefit can indeed be found in using bigger patches in image restoration – given an algorithm which can cope with the dimension increase. We present an image denoising experiment of several popular images, for increasing patch sizes. In the context of sparse representations, an image restoration task can be formulated as a Maximum a Posteriori formulation [13]. In the case of a sparse dictionary, this problem can be posed as:

$$\min_{\mathbf{z}, \mathbf{x}_i, \mathbf{A}} = \frac{\lambda}{2} \|\mathbf{z} - \mathbf{y}\|_2^2 + \sum_i \|\Phi \mathbf{A} \mathbf{x}_i - \mathbf{P}_i \mathbf{z}\|_2^2 + \mu_i \|\mathbf{x}_i\|_0, \quad (14)$$

where  $\mathbf{z}$  is the image estimate given the noisy observation  $\mathbf{y}$ ,  $\mathbf{P}_i$  is an operator that extracts the  $i^{th}$  patch from a given image and  $\mathbf{x}_i$  is the sparse representation of the  $i^{th}$  patch. We can minimize this problem by taking a similar approach to that of the dictionary learning problem: use a block-coordinate descent by fixing the unknown image  $\mathbf{z}$ , and minimizing w.r.t the sparse vectors  $\mathbf{x}_i$  and the dictionary (by any dictionary learning algorithm). We then fix the sparse vectors and update the image  $\mathbf{z}$ . Note that even though this process should be iterated (as effectively shown in [43]) we stick to the first iteration of this process to make a fair comparison with the K-SVD based algorithms.

For this experiment, denoted as Experiment 4, we use both Sparse K-SVD and OSDL, for training the double sparsity model. Each method is run with the traditional ODCT and with the cropped Wavelets dictionary, presented in this paper. We include as a reference the results of the K-SVD denoising algorithm [13], which trains a regular (dense) dictionary with patches of size

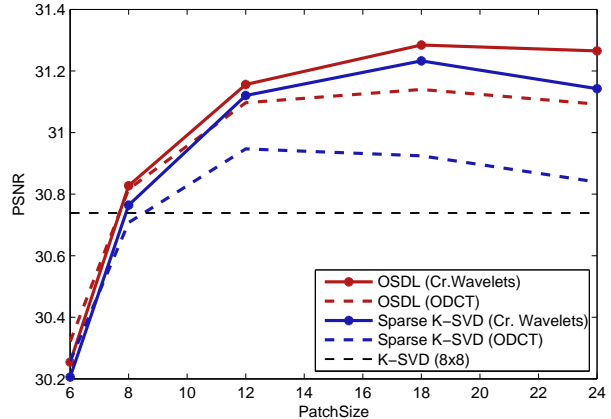


Fig. 7. Experiment 4: Denoising results as a function of the patch size for Sparse K-SVD and OSDL, which an overcomplete DCT dictionary and a separable cropped Wavelets dictionary.

$8 \times 8$ . The dictionary sparsity was set to be 10% of the signal dimension. Regarding the size of the dictionary, the redundancy was determined by the redundancy of the cropped Wavelets (as explained in Section III-A), and setting the sparse matrix  $\mathbf{A}$  to be square. This selection of parameters is certainly not optimal. For example, we could have set the redundancy as an increasing function of the signal dimension. However, learning such increasingly redundant dictionaries is limited by the finite data of each image. Therefore, we use a square matrix  $\mathbf{A}$  for all patch sizes, leaving the study of other alternatives for future work. 10 iterations were used for the K-SVD methods and 5 iterations for the OSDL.

Fig. 7 presents the averaged results over the set of 10 publicly available images used by [45], where the noise standard deviation was set to  $\sigma = 30$ . Note how the original algorithm presented in [21], Sparse K-SVD with the ODCT as the base dictionary, does not scale well with the increasing patch size. In fact, once the base dictionary is replaced by the cropped Wavelets dictionary, the same algorithm shows a jump in performance of nearly 0.4 dB. A similar effect is observed for the OSDL algorithm, where the cropped Wavelets dictionary performs the best.

Employing even greater patch sizes eventually results in decreasing denoising quality, even for the OSDL with Cropped Wavelets. Partially, this could be caused by a limitation of the sparse model in representing fine details as the dimension of the signal grows. Also, the amount of training data is limited by the size of the image, having approximately 250,000 examples to train on. Once the dimension of the patches increases, the amount of training data might become a limiting factor in the denoising performance.

As a final word about this experiment, we note that treating all patches the same way (with the same patch

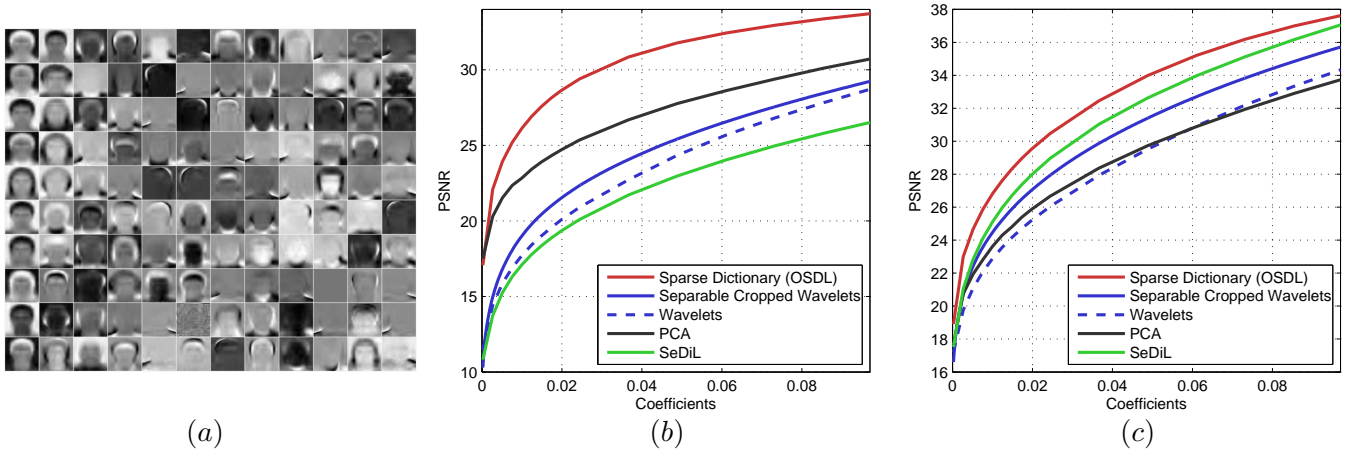


Fig. 8. Experiment 5: a) Subset of atoms from a sparse dictionary trained with OSDL on a database of aligned face images. b) Compression results (as in ratio of kept coefficients) by Wavelets, Cropped separable Wavelets, PCA, OSDL and SeDiL [18] on aligned faces. c) Compression results for the “Cropped Labeled Faces in the Wild” database.

size) is clearly not optimal. A multi-size patch approach has already been suggested in [46], though in the context of the Non-Local Means algorithm. The OSDL algorithm may be the right tool to bring multi-size patch processing to sparse representation-based algorithms, and this remains a topic of future work.

### C. Adaptive Image Compression

Image compression is the task of reducing the amount of information needed to represent an image, such that it can be stored or transmitted efficiently. In a world where image resolution increases at a surprising rate, more efficient compression algorithms are always in demand. In this section, we do not attempt to provide a complete solution to this problem but rather show how our online sparse dictionaries approach could indeed aid a compression scheme.

Most (if not all) compression methods rely on sparsifying transforms. In particular, JPEG2000, one of the best performing and popular algorithms available, is based on the 2-D Wavelet transform. Dictionary learning has already been shown to be beneficial in this application. In [47], the authors trained several dictionaries for patches of size  $15 \times 15$  on pre-aligned face pictures. These offline trained dictionaries were later used to compress images of the same type, by sparse coding the respective patches of each picture. The results reported in [47] surpass those by JPEG2000, showing the great potential of similar schemes.

In the experiment we are presenting here (Experiment 5), we go beyond the locally based compression scheme and propose to perform naive compression by just keeping a certain number of coefficients through sparse coding, where each signal is the entire target image. To this end, we use the same data set as in [47] consisting of over 11,000 examples, and re-scaled them to a size

of  $64 \times 64$ . We then train a sparse dictionary on these signals with OSDL, using the cropped Wavelets as the base dictionary for 15 iterations. For a fair comparison with other non-redundant dictionaries, in this case we chose the matrix  $\mathbf{A}$  such that the final dictionary is non-redundant (a rectangular tall matrix). A word of caution should be said regarding the relatively small training data set. Even though we are training just over 4000 atoms on only 11,000 samples, these atoms are *only* 250-sparse. This provides a great reduction to the degrees of freedom during training. A subset of the obtained atoms can be seen in Fig. 8a.

For completion, we include here the results obtained by the SeDiL algorithm [18] (with the code provided by the authors and with the suggested parameters), which trains a separable dictionary consisting of 2 small dictionaries of size  $64 \times 128$ . Note that this implies a final dictionary which has a redundancy of 4, though the degrees of freedom are of course limited due to the separability imposed.

The results of this naive compression scheme are shown in Fig. 8b for a testing set (not included in the training). As we see, the obtained dictionary performs substantially better than Wavelets – on the order of 8 dB at a given coefficient count. Partially, the performance of our method is aided by the cropped Wavelets, which in themselves perform better than the regular 2-D Wavelet transform. However, the adaptability of the matrix  $\mathbf{A}$  results in a much better compression-ratio. A substantial difference in performance is obtained after training with OSDL, even while the redundancy of the obtained dictionary is less (by about half) than the redundancy of its base-dictionary. The dictionary obtained by the SeDiL algorithm, on the other hand, has difficulties learning a completely separable dictionary for this dataset, in which the faces, despite being aligned, are difficult to



approximate through separable atoms.

As one could observe from the dictionary obtained by our method, some atoms resemble PCA-like basis elements. Therefore we include the results by compressing the testing images with a PCA transform, obtained from the same training set – essentially, performing a dimensionality reduction. As one can see, the PCA results are indeed better than Wavelets due to the regular structure of the aligned faces, but they are still relatively far from the results achieved by OSDL [2].

Lastly, we show that this naive compression scheme, based on the OSDL algorithm, does not rely on the regularity of the aligned faces in the previous database. To support this claim, we perform a similar experiment on images obtained for the “Cropped Labeled Faces in the Wild Database” [48]. This database includes images of subjects found on the web, and its *cropped* version consists of  $64 \times 64$  images including only the face of the different subjects. These face images are in different positions, orientations, resolutions and illumination conditions. We trained a dictionary for this database, which consists of just over 13,000 examples, with the same parameter as in the previous case, and the compression is evaluated on a testing set not included in the training. An analogous training process was performed with SeDiL. As shown in Fig. 8c, the PCA results are now inferior, due to the lack of regularity of the images. The separable dictionary provided by SeDiL performs better in this dataset, whose examples consists of truncated faces rather than heads, and which can be better represented by separable atoms. Yet, its representation power is compromised by its complete separability when compared to OSDL, with a 1 dB gap between the two.

#### D. Pursuing Universal Big Dictionaries

Dictionary learning has shown how to take advantage of sparse representations in specific domains, however dictionaries can also be trained for more general domains (i.e., natural images). For relatively small dimensions, several works have demonstrated that it is possible to train general dictionaries on patches extracted from non-specific natural images. Such general-purpose dictionaries have been used in many applications in image restoration, outperforming analytically-defined transforms.

Using our algorithm we want to tackle the training of such universal dictionaries for image patches of size  $32 \times 32$ , i.e., of dimension 1024. To this end, in this experiment we train a sparse dictionary with a total redundancy of 6: the cropped Wavelets dictionary introduces a redundancy of around 3, and the matrix  $\mathbf{A}$  has a redundancy of 2. The atom sparsity was set to 250, and each example was coded with 60 non-zeros in the

sparse coding stage. Training was done on 10 Million patches taken from natural images from the Berkeley Segmentation Dataset [49]. We run the OSDL algorithm for two data sweeps. For comparison, we trained a full (unconstrained) dictionary with ODL with the same redundancy, on the same database and with the same parameters.

We evaluate the quality of such a trained dictionary in an M-Term approximation experiment on 600 patches (or little images). Comparison is done with regular and separable cropped Wavelets (the last one being the base-dictionary of the double sparsity model, and as such the starting point of the training). We also want to compare our results with the approximation achieved by more sophisticated multi-scale transforms, such as Contourlets. Contourlets are a better suited multi-scale analysis for two dimensions, providing an optimal approximation rate for piece-wise smooth functions with discontinuities along twice differentiable curves [37]. This is a slightly redundant transform due to the Laplacian Pyramid used for the multi-scale decomposition (redundancy of 1.33). Note that, traditionally, hard-thresholding is used to obtain an M-term approximation, as implemented in the code made available by the authors. However, this is not optimal in the case of redundant dictionaries. We therefore construct an explicit Contourlet synthesis dictionary, and apply the same greedy pursuit we employ throughout the paper. Thus we fully leverage the approximation power of this transform, making the comparison fair<sup>8</sup>.

Moreover, and to provide a complete picture of the different transforms, we include also the results obtained for a *cropped* version of Contourlets. Since Contourlets are not separable we use a 2-D extension of our cropping procedure detailed in Section III-A to construct a cropped Contourlets synthesis dictionary. The lack of separability makes this dictionary considerably less efficient computationally. As in cropped Wavelets, we naturally obtain an even more redundant dictionary (redundancy factor of 5.3).

A subset of the obtained dictionary is shown in Fig. 9, where the atoms have been sorted according to their entropy. Very different types of atoms can be observed: from the piece-wise-constant-like atoms, to textures at

<sup>8</sup>Another option to consider is to use undecimated multi-scale transforms. The Undecimated Wavelet Transform (UDWT) [33] and the Nonsubsampled Contourlet Transform (NSCT) [50] are shift-invariant versions of the Wavelet and Contourlet transforms, respectively, and are obtained by skipping the decimation step at each scale. This greater flexibility in representation, however, comes at the cost of a huge redundancy, which becomes a prohibiting factor in any pursuing scheme. A similar undecimated scheme could be proposed for the corresponding cropped transforms, however, but this is out of the scope of this work.

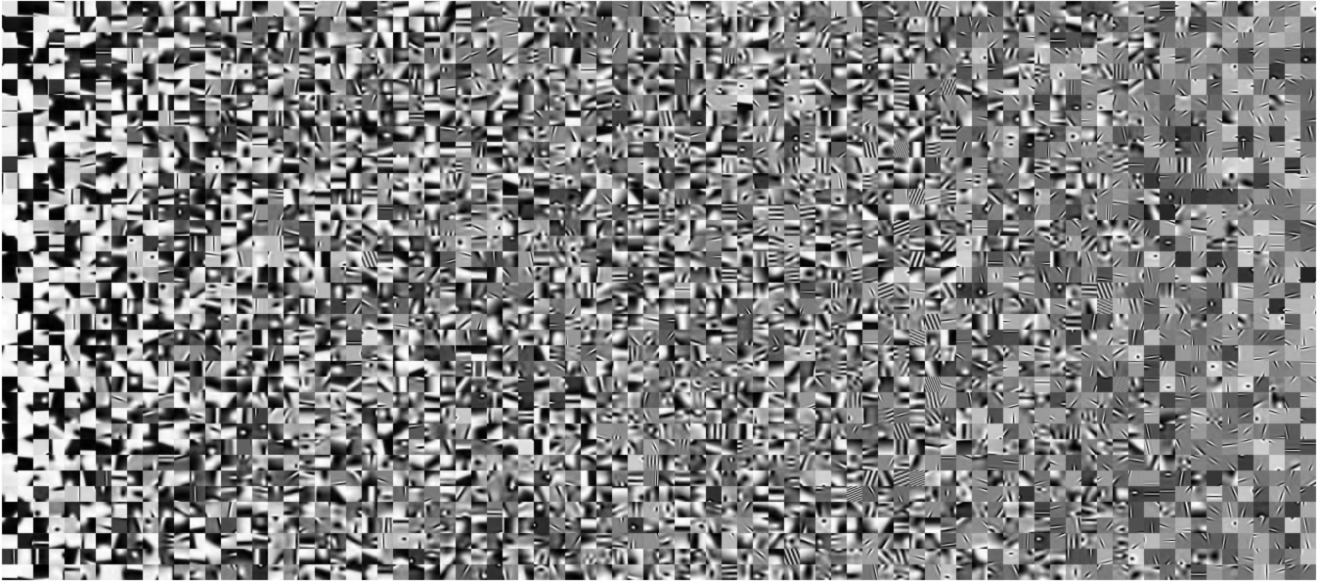


Fig. 9. Experiment 6: Subset of the general (sparse) dictionary for patches of size  $32 \times 32$  obtained with OSDL trained over 10 million patches from natural images.

different scales and edge-like atoms. It is interesting to see that Fourier type atoms, as well as Contourlet and Gabor-like atoms, naturally arise out of the training. In addition, such a dictionary obtains some flavor of shift invariance. As can be seen in Fig. 10, similar patterns may appear in different locations in different atoms. An analogous question could be posed regarding rotation invariance. Furthermore, we could consider enforcing these, or other, properties in the training. These, and many more questions, are part of on-going work.

The approximation results are shown in Fig. 11.a, where Contourlets can be seen to perform slightly better than Wavelets. The cropping of the atoms significantly enhances the results for both transforms, with a slight advantage for cropped Wavelets over cropped Contourlets. The Trainlets, obtained with OSDL, give the highest PSNR. Interestingly, the ODL algorithm by [6] performs slightly worse than the proposed OSDL, despite the vast database of examples. In addition, the learning (two epochs) with ODL took roughly 4.6 days, whereas the OSDL took approximately 2 days<sup>9</sup>. As we see, the sparse structure of the dictionary is not only beneficial in cases with limited training data (as in Experiment 1), but also in this big data scenario. We conjecture that this is due to the better guiding of the training process, helping to avoid local minima which an unconstrained dictionary might be prone to.

As a last experiment, we want to show that our scheme can be employed to train an adaptive dictionary for even

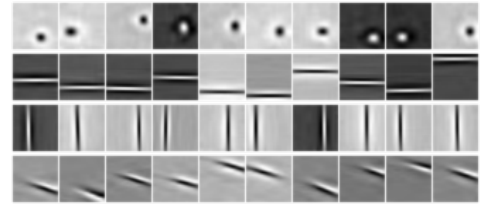


Fig. 10. Experiment 6: Atoms of size  $32 \times 32$  with recurring patterns at different locations.

higher dimensional signals. In Experiment 8, we perform a similar training with OSDL on patches (or images) of size  $64 \times 64$ , using an atom sparsity of 600. The cropped Wavelets dictionary has a redundancy of 2.44, and we set  $\mathbf{A}$  to be square. For a fair comparison, and due to the extensive time involved in running ODL, we first ran ODL for 5 days, giving it sufficient time for convergence. During this time ODL accessed 3.8 million training examples. We then ran OSDL using the same examples<sup>10</sup>.

As shown in Fig. 11.b, the relative performance of the different methods is similar to the previous case. Trainlets again gives the best approximation performance, giving a glimpse into the potential gains achievable when training can be effectively done at larger signal scales. It is not possible to show here the complete trained dictionary, but we do include some selected atoms from it in Fig. 11.c. We obtain many different types of atoms: from the very local curvelets-like atoms, to more global Fourier atoms, and more.

<sup>10</sup>The provided code for ODL is not particularly well suited for cluster-processing (needed for this experiment), and so the times involved in this case should not be taken as an accurate run-time comparison.

<sup>9</sup>This experiment was run on a 64-bit operating system with an Intel Core i7 microprocessor, with 16 Gb of RAM, in Matlab.

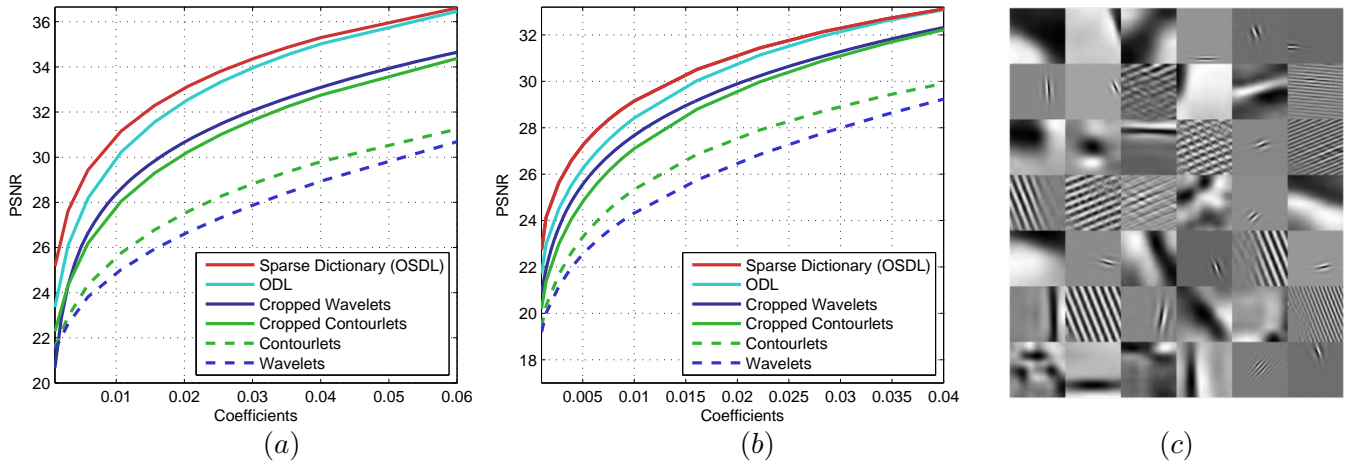


Fig. 11. Experiment 7-8: a) M-term approximation of general image patches of size 32x32 for different methods. b) M-term approximation of general image patches of size 64x64 for different methods. c) Some atoms of size  $64 \times 64$  from the dictionary trained with OSDL.

## VI. SUMMARY AND FUTURE WORK

This work shows that dictionary learning can be up-scaled to tackle a new level of signal dimensions. We propose a modification on the Wavelet transform by constructing two-dimensional separable *cropped* Wavelets, which allow a multi-scale decomposition of patches without significant border effects. We apply these Wavelets as a base-dictionary within the Double Sparsity model, allowing this approach to now handle larger and larger signals. In order to handle the vast data sets needed to train such a big model, we propose an Online Sparse Dictionary Learning algorithm, employing SGD ideas in the dictionary learning task. We show how, using these methods, dictionary learning is no longer limited to small signals, and can now be applied to obtain Trainlets, high dimensional trainable atoms.

While OMP proved sufficient for the experiments shown in this work, considering other sparse coding algorithms might be beneficial. In addition, the entire learning algorithm was developed using a strict  $l_0$  pseudo-norm, and its relaxation to other convex norms opens new possibilities in terms of training methods. Another direction is to extend our model to allow for the adaptability of the separable base-dictionary itself, incorporating ideas of separable dictionary learning thus providing a completely adaptable structure. Understanding quantitatively how different parameters affect the learned dictionaries, such as redundancy and atom sparsity, will provide a better understanding of our model. These questions, among others, are part of ongoing work.

## VII. ACKNOWLEDGEMENTS

The research leading to these results has received funding from the European Research Council under European Unions Seventh Framework Programme, ERC

Grant agreement no. 320649. This research was supported by Israel Science Foundation (ISF) grant number 1770/14. The authors would like to thank the anonymous reviewers who helped improve the quality of this manuscript, as well as the authors of [18] for generously providing their code and advice for comparison purposes.

## REFERENCES

- [1] M. Elad, *Sparse and Redundant Representations: From Theory to Applications in Signal and Image Processing*. Springer Publishing Company, Incorporated, 1st ed., 2010.
- [2] A. M. Bruckstein, D. L. Donoho, and M. Elad, "From Sparse Solutions of Systems of Equations to Sparse Modeling of Signals and Images," *SIAM Review*, vol. 51, pp. 34–81, Feb. 2009.
- [3] R. Rubinstein, A. M. Bruckstein, and M. Elad, "Dictionaries for sparse representation modeling," *IEEE Proceedings - Special Issue on Applications of Sparse Representation & Compressive Sensing*, vol. 98, no. 6, pp. 1045–1057, 2010.
- [4] M. Aharon, M. Elad, and A. M. Bruckstein, "K-SVD: An Algorithm for Designing Overcomplete Dictionaries for Sparse Representation," *IEEE Trans. on Signal Process.*, vol. 54, no. 11, pp. 4311–4322, 2006.
- [5] K. Engan, S. O. Aase, and J. H. Husoy, "Method of Optimal Directions for Frame Design," in *IEEE Int. Conf. Acoust. Speech, Signal Process.*, pp. 2443–2446, 1999.
- [6] J. Mairal, F. Bach, J. Ponce, and G. Sapiro, "Online Learning for Matrix Factorization and Sparse Coding," *J. Mach. Learn. Res.*, vol. 11, pp. 19–60, 2010.
- [7] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian, "Image Denoising by Sparse 3-D Transform-Domain Collaborative Filtering," *IEEE Trans. on Image Process.*, vol. 16, pp. 2080–2095, Jan. 2007.
- [8] J. Mairal, F. Bach, and G. Sapiro, "Non-local Sparse Models for Image Restoration," *IEEE International Conference on Computer Vision*, vol. 2, pp. 2272–2279, 2009.
- [9] D. Zoran and Y. Weiss, "From learning models of natural image patches to whole image restoration," *2011 International Conference on Computer Vision, ICCV*, pp. 479–486, Nov. 2011.
- [10] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian, "Image denoising with block-matching and 3D filtering," *Proc. SPIE-IS&T Electron. Imaging*, vol. 6064, pp. 1–12, 2006.

- [11] W. Dong, L. Zhang, G. Shi, and X. Wu, "Image deblurring and super-resolution by adaptive sparse domain selection and adaptive regularization," *IEEE Trans. on Image Process.*, vol. 20, no. 7, pp. 1838–1857, 2011.
- [12] J. Yang, J. Wright, T. S. Huang, and Y. Ma, "Image super-resolution via sparse representation," *IEEE Trans. on Image Process.*, vol. 19, no. 11, pp. 2861–2873, 2010.
- [13] M. Elad and M. Aharon, "Image denoising via sparse and redundant representations over learned dictionaries," *IEEE Trans. Image Process.*, vol. 15, pp. 3736–3745, Dec. 2006.
- [14] Y. Romano, M. Protter, and M. Elad, "Single image interpolation via adaptive nonlocal sparsity-based modeling," *IEEE Trans. on Image Process.*, vol. 23, no. 7, pp. 3085–3098, 2014.
- [15] B. Ophir, M. Lustig, and M. Elad, "Multi-Scale Dictionary Learning Using Wavelets," *IEEE J. Sel. Top. Signal Process.*, vol. 5, pp. 1014–1024, Sept. 2011.
- [16] J. Sulam, B. Ophir, and M. Elad, "Image Denoising Through Multi-Scale Learnt Dictionaries," in *IEEE International Conference on Image Processing*, pp. 808 – 812, 2014.
- [17] H. C. Burger, C. J. Schuler, and S. Harmeling, "Image denoising: Can plain neural networks compete with BM3D?," *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2392–2399, 2012.
- [18] S. Hawe, M. Seibert, and M. Kleinsteuber, "Separable dictionary learning," *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 438–445, 2013.
- [19] M. Aharon and M. Elad, "Sparse and Redundant Modeling of Image Content Using an Image-Signature-Dictionary," *SIAM Journal on Imaging Sciences*, vol. 1, no. 3, pp. 228–247, 2008.
- [20] L. Benoit, J. Mairal, F. Bach, and J. Ponce, "Sparse image representation with epitomes," in *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pp. 2913–2920, IEEE, 2011.
- [21] R. Rubinstein, M. Zibulevsky, and M. Elad, "Double Sparsity : Learning Sparse Dictionaries for Sparse Signal Approximation," *IEEE Trans. Signal Process.*, vol. 58, no. 3, pp. 1553–1564, 2010.
- [22] M. Yaghoobi and E. Davies, Mike, "Compressible dictionary learning for fast sparse approximations," in *IEEE/SP 15th Workshop on Statistical Signal Processing*, pp. 662–665, Aug. 2009.
- [23] L. Le Magoarou and R. Gribonval, "Chasing butterflies: In search of efficient dictionaries," in *IEEE Int. Conf. Acoust. Speech, Signal Process.*, Apr. 2015.
- [24] O. Chabiron, F. Malgouyres, J. Tourneret, and N. Dobigeon, "Toward Fast Transform Learning," *International Journal of Computer Vision*, pp. 1–28, 2015.
- [25] M. Elad, P. Milanfar, and R. Rubinstein, "Analysis versus synthesis in signal priors," *Inverse Problems*, vol. 23, pp. 947–968, 2007.
- [26] R. Rubinstein and M. Elad, "Dictionary Learning for Analysis-Synthesis Thresholding," *IEEE Trans. on Signal Process.*, vol. 62, no. 22, pp. 5962–5972, 2014.
- [27] S. Ravishanker and Y. Bresler, "Learning Sparsifying Transforms," *IEEE Trans. Signal Process.*, vol. 61, no. 5, p. 61801, 2013.
- [28] S. Ravishanker, B. Wen, and Y. Bresler, "Online Sparsifying Transform Learning Part I: Algorithms," *IEEE J. Sel. Top. Sig. Proc.*, vol. 9, no. 4, pp. 625–636, 2015.
- [29] S. Ravishanker and Y. Bresler, "Learning doubly sparse transforms for images," *IEEE Trans. Image Process.*, vol. 22, no. 12, pp. 4598–4612, 2013.
- [30] L. Bottou, "Online algorithms and stochastic approximations," in *Online Learning and Neural Networks*, Cambridge University Press, 1998. revised, Oct 2012.
- [31] M. Almeida and M. Figueiredo, "Frame-based image deblurring with unknown boundary conditions using the alternating direction method of multipliers," in *IEEE International Conference on Image Processing (ICIP)*, pp. 582–585, Sept 2013.
- [32] S. Reeves, "Fast image restoration without boundary artifacts," *IEEE Trans. Image Process.*, vol. 14, pp. 1448–1453, Oct 2005.
- [33] S. Mallat, *A Wavelet Tour of Signal Processing, Third Edition: The Sparse Way*. Academic Press, 3rd ed., 2008.
- [34] A. Cohen, I. Daubechies, and P. Vial, "Wavelet bases on the interval and fast algorithms," *Journal of Applied and Computational Harmonic Analysis*, vol. 1, no. 12, pp. 54–81, 1993.
- [35] Y. Zhao and D. Malah, "Improved segmentation and extrapolation for block-based shape-adaptive image coding," in *Proc. Vision Interface*, pp. 388–394, 2000.
- [36] E. J. Candes and D. L. Donoho, "Curvelets, multiresolution representation, and scaling laws," in *Proc. SPIE*, vol. 4119, pp. 1–12, 2000.
- [37] M. N. Do and M. Vetterli, "The contourlet transform: an efficient directional multiresolution image representation," *IEEE Trans. Image Process.*, vol. 14, no. 12, pp. 2091–2106, 2005.
- [38] T. Blumensath and M. E. Davies, "Normalized iterative hard thresholding: Guaranteed stability and performance," *IEEE Journal on Selected Topics in Signal Processing*, vol. 4, no. 2, pp. 298–309, 2010.
- [39] T. Blumensath and M. E. Davies, "Iterative Thresholding for Sparse Approximations," *Journal of Fourier Analysis and Applications*, vol. 14, pp. 629–654, Sept. 2008.
- [40] L. Bottou and O. Bousquet, "The Tradeoffs of Large Scale Learning," *Artificial Intelligence*, vol. 20, pp. 161–168, 2008.
- [41] R. Rubinstein, M. Zibulevsky, and M. Elad, "Efficient Implementation of the K-SVD Algorithm using Batch Orthogonal Matching Pursuit," *Technion - Computer Science Department - Technical Report.*, pp. 1–15, 2008.
- [42] J. Mairal, F. Bach, J. Ponce, and G. Sapiro, "Online dictionary learning for sparse coding," in *International Conf. on Machine Learning*, pp. 689–696, ACM, 2009.
- [43] J. Sulam and M. Elad, "Expected patch log likelihood with a sparse prior," in *Energy Minimization Methods in Computer Vision and Pattern Recognition*, Lecture Notes in Computer Science, pp. 99–111, Springer International Publishing, 2015.
- [44] Y. Romano and M. Elad, "Boosting of Image Denoising Algorithms," *SIAM Journal on Imaging Sciences*, vol. 8, no. 2, pp. 1187–1219, 2015.
- [45] M. Lebrun, A. Buades, and J.-M. Morel, "Implementation of the non-local bayes(nl-bayes) image denoising algorithm," *Image Processing On Line*, vol. 2013, pp. 1–42, 2013.
- [46] A. Levin, B. Nadler, F. Durand, and W. T. Freeman, "Patch complexity, finite pixel correlations and optimal denoising," in *Computer Vision–ECCV 2012*, pp. 73–86, Springer, 2012.
- [47] O. Bryt and M. Elad, "Compression of facial images using the K-SVD algorithm," *J. Vis. Commun. Image Represent.*, vol. 19, pp. 270–282, May 2008.
- [48] C. Sanderson and B. C. Lovell, "Multi-region probabilistic histograms for robust and scalable identity inference," *Lecture Notes in Computer Science*, pp. 199–208, 2009.
- [49] D. Martin, C. Fowlkes, D. Tal, and 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*, vol. 2, pp. 416–423, July 2001.
- [50] R. Eslami and H. Radha, "Translation-invariant contourlet transform and its application to image denoising," *IEEE Trans. Image Process.*, vol. 15, no. 11, pp. 3362–3374, 2006.