SIAM J. IMAGING SCIENCES
Vol. 1, No. 3, pp. 228–247

Sparse and Redundant Modeling of Image Content Using an Image-Signature-Dictionary*

Michal Aharon† and Michael Elad†

Abstract. Modeling signals by sparse and redundant representations has been drawing considerable attention in recent years. Coupled with the ability to train the dictionary using signal examples, these techniques have been shown to lead to state-of-the-art results in a series of recent applications. In this paper we propose a novel structure of such a model for representing image content. The new dictionary is itself a small image, such that every patch in it (in varying location and size) is a possible atom in the representation. We refer to this as the *image-signature-dictionary* (ISD) and show how it can be trained from image examples. This structure extends the well-known image and video epitomes, as introduced by Jojic, Frey, and Kannan [in *Proceedings of the IEEE International Conference on Computer Vision*, 2003, pp. 34–41] and Cheung, Frey, and Jojic [in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2005, pp. 42–49], by replacing a probabilistic averaging of patches with their sparse representations. The ISD enjoys several important features, such as shift and scale flexibilities, and smaller memory and computational requirements, compared to the classical dictionary approach. As a demonstration of these benefits, we present high-quality image denoising results based on this new model.

Key words. sparse representation, matching pursuit, denoising, image-signature, dictionary, learning, MOD

AMS subject classifications. 68U10, 62H35

DOI. 10.1137/07070156X

1. Introduction.

1.1. Background.

In sparse and redundant modeling of signals, a signal $\mathbf{y} \in \mathbb{R}^n$ is represented as a linear combination of a few prototype signals taken from a *dictionary* $\mathbf{D} \in \mathbb{R}^{n\times m}$. This dictionary contains a collection of m atoms $\mathbf{d}_i \in \mathbb{R}^n$ that are the building blocks of the representation. A representation of the signal $\mathbf{y}$ is then any vector $\mathbf{x} \in \mathbb{R}^m$ satisfying $\mathbf{y} = \mathbf{D}\mathbf{x}$.

In the case where $m > n$, the representation $\mathbf{x}$ becomes redundant, and there are infinitely many possible solutions to the system $\mathbf{y} = \mathbf{D}\mathbf{x}$. Among this infinitely large set of solutions, the sparsest one is preferred, i.e., the one with the smallest $\|\mathbf{x}\|_0$-norm[1] [12, 24, 20, 41]. Thus, the task of computing a representation for a signal can be formally described by

$$\min_{\mathbf{x}} \|\mathbf{x}\|_0 \quad \text{subject to} \quad \mathbf{y} = \mathbf{D}\mathbf{x}. \tag{1}$$

As often happens in this field, the exact equality in the constraint above is relaxed and

*Received by the editors August 30, 2007; accepted for publication (in revised form) April 9, 2008; published electronically July 30, 2008.
http://www.siam.org/journals/siims/1-3/70156.html
†Computer Science Department, Technion–Israel Institute of Technology, Technion City, Haifa 32000, Israel (michalo@cs.technion.ac.il, elad@cs.technion.ac.il).
[1]The notation $\|\mathbf{x}\|_0$ stands for a count of the number of nonzeros in the vector $\mathbf{x}$. While this is not a formal norm due to the violation of the homogeneity property, it is related to the regular ℓ^p-norm by $p \to 0$.

replaced by the alternative requirement $\|\mathbf{y} - \mathbf{D}\mathbf{x}\|_2 \leq \epsilon$, to allow for additive noise and model deviations.

Solving this problem was proved to be an NP-hard problem [33]. However, many approximation techniques for this task were proposed. Two common methods are matching-pursuit (MP) methods [32, 35, 41] that find the solution one entry at a time in a greedy way, and the basis-pursuit (BP) [22, 7] algorithm that replaces the ℓ^0- by the ℓ^1-norm. A wide theoretical study has shown that if a sparse enough solution exists, it will be found by these approximation techniques [12, 13, 14, 42, 21, 23].

A key question in a successful deployment of the above model to signals is the choice of dictionary. Prespecified dictionaries, based on known transforms and their variations, have been used extensively in the past decade with varying degrees of success. These include various forms of wavelets and wavelet packets [9], steerable wavelets [36], curvelets [40], contourlets [11], and bandelets [31].

An alternative approach is one of training: a design of the dictionary to efficiently represent a given set of signals by a learning procedure. In this approach, a dictionary $\mathbf{D}$ of size $n \times m$ is built such that it leads to the sparsest representation for a set of training signals. A pioneering work by Olshausen and Field set the stage for such techniques [34], to be followed later by a sequence of contributions, among which we mention [28, 17, 2, 3, 18]. In all these methods, an example set of training signals is gathered. These examples are expected to be of similar nature to the signals to be operated upon. After initializing the dictionary, these methods iterate between sparse representation of the training signals based on the current dictionary, and updating the dictionary to improve the representations. It was shown in [2, 3] that such an approach can be thought of as a generalization of the K-means algorithm for the vector quantization problem, which addresses the limited case of finding the best dictionary for representation of signals by one atom only and with coefficients limited to identity.

The combination of sparse and redundant representation modeling of signals, together with a learned dictionary, proves its superiority in various applications in image processing. State-of-the-art results are obtained in texture classification [39], denoising of still images and video [15, 16, 38], color image inpainting and demosaicing [30], and more. A multiscale version of the learning process and various other applications in signal and image processing are topics of current study.

1.2. This paper. In this work we adopt the above-described dictionary learning approach and apply it with a novel structure that enjoys several important features, such as shift and scale flexibilities and smaller memory and computational requirements, compared to the classical dictionary approach. This novel structure, referred to hereafter as the *image-signature-dictionary* (ISD), is an image (two-dimensional array of scalar values) in which each patch can serve as a representing atom. As such, a near shift-invariant property is obtained, due to the overlap between atoms extracted from the ISD in nearby locations. Similarly, by taking patches of varying sizes, near scale-invariance is potentially obtained and exploited.

The idea that a small image could be learned and used to represent a larger one has already been proposed in recent years. The smaller representing image has been called the *epitome*. Learning epitomes for images, and later for video, were presented originally in [26, 8] and later extended in [27, 29]. In this representation, patches from the image are found and

extracted from the epitome. This idea was shown to effectively represent visual content and enable various applications [26, 8, 27, 29]. The ISD presented in this paper is a generalized version of the epitome idea, in the sense that the simpler epitome direct representation of the patches is replaced by a more flexible sparse representation modeling. As we shall show in the experimental part of this paper, this difference is crucial in getting state-of-the-art denoising performance. Furthermore, the ISD training algorithm developed in this paper is far simpler than the method for learning epitomes, as presented in [26, 8]. We shall expand on these differences in sections 2 and 3.

In the next section we introduce the proposed structure and compare its features with those of a regular dictionary and epitomes. Section 3 is devoted to the development of a training algorithm for the ISD, presenting both the algorithm and illustrative experimental results. Section 4 presents a detailed image denoising algorithm relying on the ISD, exploiting its special structure to gain both in computations and in output quality. We conclude the paper in section 5 with a summary of this work and a description of its possible future extensions.

2. ISD: The proposed structure. Let $\mathbf{D}_S \in \mathbb{R}^{\sqrt{m}\times\sqrt{m}}$ be such a signature dictionary, and assume that an image patch $\mathbf{y}_\square \in \mathbb{R}^{\sqrt{n}\times\sqrt{n}}$ is to be represented by its atoms. Here and throughout the paper we use square arrays for both the ISD and the patches extracted from it as atoms. This is done for simplicity of notation, as nothing in the following developments restricts the choice to squares. We define $\mathbf{d}_S \in \mathbb{R}^m$ as a vector obtained by a column-lexicographic ordering of the ISD, and, similarly, $\mathbf{y} \in \mathbb{R}^n$ as a vector representing the given image patch as a single column.

Define $\mathbf{C}_{[k,l]} \in \mathbb{R}^{n\times m}$ as the linear operator that extracts a patch of size $\sqrt{n}\times\sqrt{n}$ from the ISD $\mathbf{D}_S$ in location (top left corner, say) $[k,l]$; i.e., $\mathbf{C}_{[k,l]}\mathbf{d}_S$ is the extracted patch as a column vector. We shall assume hereafter that such an extraction is a cyclic operation, implying that when the chosen patch rolls over the right side of the array or its bottom, it proceeds with the assumption that the ISD is periodic.

Using the above, $\mathbf{y}$ can be represented as a linear combination of patches (atoms) of the same size, taken at various locations in $\mathbf{D}_S$:

$$\mathbf{y} = \sum_{k=1}^{\sqrt{m}}\sum_{l=1}^{\sqrt{m}} x_{[k,l]}\mathbf{C}_{[k,l]}\mathbf{d}_S. \tag{2}$$

This construction could be reformulated by extracting all m possible patches of size $\sqrt{n}\times\sqrt{n}$ from $\mathbf{D}_S$ and converting them to vectors (this can be achieved by computing $\mathbf{C}_{[k,l]}\mathbf{d}_S$ for $1 \le k,l \le \sqrt{m}$) and gathering these vectors as columns of a matrix $\mathbf{D} \in \mathbb{R}^{n\times m}$. Once constructed, the representation shown in (2) becomes identical to the one posed earlier, namely,

$$\mathbf{y} = \sum_{k=1}^{\sqrt{m}}\sum_{l=1}^{\sqrt{m}} x_{[k,l]}\mathbf{C}_{[k,l]}\mathbf{d}_S = \mathbf{D}\mathbf{x}. \tag{3}$$

We see that for the sizes chosen, the number of atoms in the dictionary is the same (m) in the two methods—the original and the signature dictionaries. However, there is a marked difference—whereas the number of free parameters in a regular dictionary is mn, here it is only

m. Indeed, the ISD atoms are highly constrained, as each atom resembles others extracted from nearby locations. Such overlap between the atoms is natural for representation of image patches, as it is tightly coupled with the idea of the shift-invariance property, although in a different way from the one discussed in [6, 18, 1].

The fact that fewer parameters are involved in the definition of the ISD means that a smaller number of computations could potentially be used for its training and use in applications. On the other hand, it is also clear that with a reduced number of degrees of freedom, the flexibility of such a model to describe natural data is more limited. However, since shift-invariance (or rather, affinity to it) is a property we expect natural images to benefit from, we could hope that the new structure brings an overall benefit. Experimental results to be presented hereafter confirm this hope in various ways.

As opposed to a regular dictionary, a single ISD can represent image patches with varying sizes, simply by updating the operators $\mathbf{C}_{[k,l]}$ to extract patches of the desired size. In fact, if we desire to use the ISD for several scales, it should be trained as such. We shall present this option in details in the next section. Figure 1 demonstrates the ISD and the flexibilities it introduces.

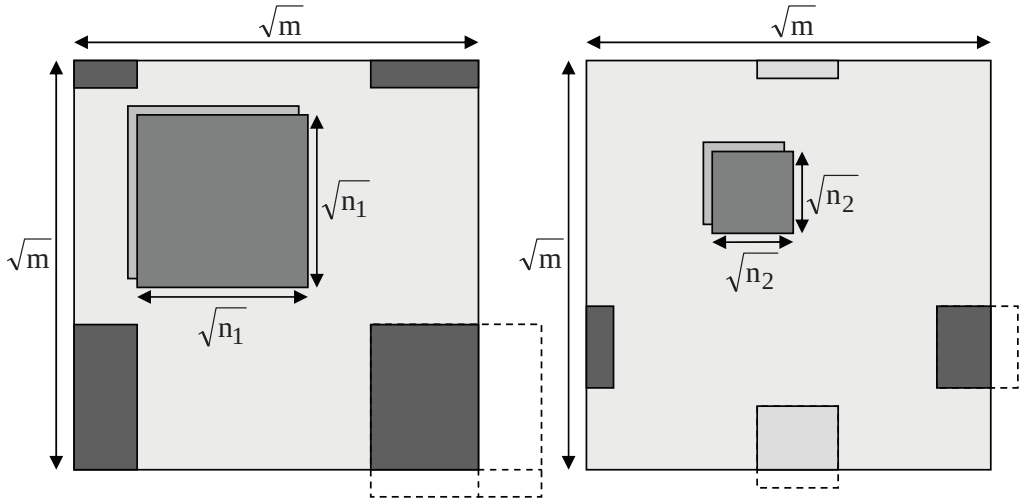

Figure 1. *Illustration of the ISD, the ability to extract atoms of varying sizes from it, and the shift and cyclic nature of the atoms taken from it.*

As mentioned in the introduction, the idea of using a small image to represent a larger one has been already proposed in recent years by Jojic, Frey, and Kannen [26] and Cheung, Frey, and Jojic [8]. Their representative image, coined *epitome*, is trained from the input image, just as we intend to do with the ISD. The image can be built by combining patches from the epitome in a probabilistic averaging manner. Indeed, the work presented in [26, 8], and later extended in [27, 29], does much more than that, as it includes the ability to segment the image into different regions, provide an alpha blending map for the fusion of several layers, and build an epitome for this part (the shape) as well. This treatment enables the use of epitomes for separating an image into layers, inpainting these layers beyond their existing

supports, denoising the image/video, superresolving it, and more.

In our work we restrict our treatment to what is referred to in [26] as the texture or the appearance epitome, which removes the segmentation part, considering the entire image as a whole. While the epitome is used to construct the desired image by directly copying patches to the image, the ISD will do this construction by linearly combining several carefully chosen patches. Thus, the ISD can be referred to as a generalization of the texture epitome, allowing the representation to use several patches, instead of only one, and enabling their fusion by simple weighting. Indeed, the relation between the epitome and the ISD parallels the one between vector quantization (VQ), which is a clustering dictionary, and a regular sparse representation dictionary. Whereas the VQ represents every data instance as the closest member in the VQ dictionary, the sparse representation modeling allows the use of a small set of atoms. At the extreme, when the ISD is forced to use only one atom per patch, and when the coefficient is constrained to be 1, the ISD and the texture epitome coincide.

Due to the above reasoning, one might expect the ISD training and usage algorithms to be far more complex, when compared to those developed for the epitomes in [26]. As we present next, the proposed algorithms are in fact far simpler, due to the natural relation to the recently studied problem of learning sparse representation dictionaries.

3. ISD training and usage.

3.1. Formulating the problem for the single scale. Given a set of training patches $\{\mathbf{y}_{\square}^{i}\}_{i=1}^{N} \in \mathbb{R}^{\sqrt{n}\times\sqrt{n}}$ (or their vector representations $\{\mathbf{y}_i\}_{i=1}^{N}$), we search for an ISD $\mathbf{D}_S \in \mathbb{R}^{\sqrt{m}\times\sqrt{m}}$ that leads to the best (i.e., sparsest) representation of all these patches. For simplicity we shall assume that each patch is represented by a fixed number of atoms,[2] L. We start by defining the energy function that the ISD is expected to minimize. Similarly to the way it has been defined in [28, 17, 2, 3, 18], we write

$$(4)\quad \hat{\mathbf{d}}_S = \operatorname{Arg}\min_{\mathbf{d}_S} \sum_{i=1}^{N} \epsilon_i^2(\mathbf{d}_S) \quad \text{subject to}$$

$$\epsilon_i^2(\mathbf{d}_S) = \min_{\mathbf{x}} \left\| \mathbf{y}_i - \sum_{k=1}^{\sqrt{m}}\sum_{l=1}^{\sqrt{m}} x_{[k,l]}\mathbf{C}_{[k,l]}\mathbf{d}_S \right\|_2^2 \quad \text{subject to} \quad \|\mathbf{x}\|_0 \le L,\ 1 \le i \le N,$$

where the vector $\mathbf{x}$ is simply the concatenation of the m coefficients in the array $x_{[k,l]}$. This problem is a *bi-level optimization* (BLO) problem [4]: per any chosen signature dictionary $\mathbf{d}_S$, the inner part seeks a sparse enough representation (with L atoms at the most) that leads to minimal representation error for each of the examples. This defines a representation error $\epsilon_i^2(\mathbf{d}_S)$. Among all possibilities of $\mathbf{d}_S$ we desire the one with the smallest accumulated such error.

The approach we take for solving this problem is an iterated and interlaced update of the ISD and the sparse representations. Given the current value of $\mathbf{d}_S$ we can solve the inner optimization problem and find the sparse representation for each example—we refer to

[2] Changing this to representations with a flexible number of atoms, driven by an error threshold, is quite simple.

this stage as the *sparse coding* stage. Once found, we can consider these representations as fixed and solve the outer optimization, forming the *dictionary update stage*. This structure should remind the reader of the K-means algorithm for clustering and more recent methods for sparse representation dictionary learning [28, 17, 2, 3]. We emphasize that while this overall algorithm does converge to a fixed point, one cannot guarantee a global minimum of the energy function in (4), due to the nonconvexity of this function.

Before turning to discussing the details of this algorithm, we draw the reader's attention again to the epitomes, as found in [26, 8]. Learning the texture-epitome for a single image (or video) is also built as an energy minimization task. However, the energy function proposed leans strongly on more complex statistical modeling of the epitome patches, the input image patches, and the ties between them, defining a larger set of unknowns to be formed. These unknowns include the texture-epitome itself, the variance for each pixel in it, and the distribution of the epitome patches. As we show next, the method developed here is far simpler, and yet it leads to a highly effective signature representation.

3.2. The sparse coding stage. Assuming that $\mathbf{d}_S$ is known and fixed, the inner optimization task in (4) reads

$$\hat{\mathbf{x}}_i = \operatorname*{Arg\,min}_{\mathbf{x}} \left\| \mathbf{y}_i - \sum_{k=1}^{\sqrt{m}} \sum_{l=1}^{\sqrt{m}} x_{[k,l]} \mathbf{C}_{[k,l]} \mathbf{d}_S \right\|_2^2 \quad \text{subject to} \quad \|\mathbf{x}\|_0 \leq L, \tag{5}$$

which is equivalent to

$$\hat{\mathbf{x}}_i = \operatorname*{Arg\,min}_{\mathbf{x}} \|\mathbf{y}_i - \mathbf{D}\mathbf{x}\|_2^2 \quad \text{subject to} \quad \|\mathbf{x}\|_0 \leq L, \tag{6}$$

where $\mathbf{D}$ is the accumulation of all m atoms in the ISD.

The solution of this problem can be obtained by any pursuit algorithm, as mentioned in the introduction. In this work we use a greedy method—the orthogonal matching pursuit (OMP)—because of its relative simplicity and efficiency [32, 35, 41]. The OMP selects at each stage an atom from the dictionary that best resembles the residual. After each such selection, the signal is back-projected onto the set of chosen atoms, and the new residual signal is calculated.

A direct use of the OMP with the ISD could be proposed by converting $\mathbf{D}_S$ (or $\mathbf{d}_S$) to a regular dictionary format $\mathbf{D} \in \mathbb{R}^{n \times m}$, as described above. However, in such a format, we do not exploit the overlap between the atoms. Considering this fact, the OMP and other pursuit techniques can be accelerated. For example, all projections between a signal and all dictionary atoms can be computed by only one inner product between the Fourier transforms of the signal and the ISD, exploiting the equivalence between a convolution in the spatial domain and an inner product in the frequency domain [37].

Thus, while a straightforward projection of a signal onto all possible atoms is done in $\mathcal{O}(mn)$ flops, the same result can be achieved in $\mathcal{O}(m \log m)$ flops, when using forward and inverse Fourier transforms of the signal. Moreover, this bound can be reduced to $O(m)$ per signal, when the dictionary is assumed fixed and its Fourier transform is computed offline beforehand. Such an assumption is reasonable when applying the same dictionary in many

examples, as happens in the training process. A similar savings in computations is presented in [19], although in a different context, where an image is convolved with an atom from a dictionary.

A side benefit to the above discussion is the fact that one can use larger atoms than with the regular dictionary structure, while keeping the overall complexity reasonably low.

3.3. Dictionary update stage. Assuming now that the sparse representation vectors $\hat{\mathbf{x}}_i$ have been computed, we seek the best ISD $\mathbf{d}_S$ to minimize

$$E(\mathbf{d}_S) = \sum_{i=1}^{N} \left\| \mathbf{y}_i - \sum_{k=1}^{\sqrt{m}} \sum_{l=1}^{\sqrt{m}} x^i_{[k,l]} \mathbf{C}_{[k,l]} \mathbf{d}_S \right\|_2^2. \tag{7}$$

This is a simple quadratic expression with respect to $\mathbf{d}_S$. As such, the update of the ISD can be performed in one step by solving a set of m linear equations, similar to the way the method of optimal directions (MOD) algorithm operates [28, 17, 18]. The gradient of the error expression in (7) is given by

$$\frac{\partial E(\mathbf{d}_S)}{\partial \mathbf{d}_S} = \mathbf{R}\mathbf{d}_S - \mathbf{p}, \tag{8}$$

where we have defined

$$\mathbf{R} = \sum_{i=1}^{N} \left(\sum_{k=1}^{\sqrt{m}} \sum_{l=1}^{\sqrt{m}} x^i_{[k,l]} \mathbf{C}_{[k,l]} \right)^T \left(\sum_{k=1}^{\sqrt{m}} \sum_{l=1}^{\sqrt{m}} x^i_{[k,l]} \mathbf{C}_{[k,l]} \right) \in \mathbb{R}^{m \times m} \tag{9}$$

and

$$\mathbf{p} = \sum_{i=1}^{N} \left(\sum_{k=1}^{\sqrt{m}} \sum_{l=1}^{\sqrt{m}} x^i_{[k,l]} \mathbf{C}_{[k,l]} \right)^T \mathbf{y}_i \in \mathbb{R}^m. \tag{10}$$

Thus, the optimal ISD is obtained simply by

$$\hat{\mathbf{d}}_S = \mathbf{R}^{-1}\mathbf{p}. \tag{11}$$

Assuming that only L entries of the $\mathbf{x}_i$ vectors are nonzeros, the summations in the definition of $\mathbf{R}$ and $\mathbf{p}$ include NL terms, which is far smaller than Nm. For moderate sizes of m (up to $m = 1000$) one could solve the above directly. For a larger ISD, an iterative solver, such as a conjugate gradient, could be proposed.

Each dictionary update stage promises to reduce the overall representation error. If we assume that the sparse coding stage is also successful in finding the smallest representation error for the given sparsity, the overall algorithm necessarily converges. In cases where OMP fails to find a better approximation, this can be easily detected (by comparing the result to the previous one), and then the solution is simply chosen as the previous one. This guarantees a monotonic decrease of the representation error for the set of training examples.

3.4. Stochastic gradient approach. The training algorithm described above updates the sparse representations $\mathbf{x}_i$ for all the examples $\{\mathbf{y}_i\}_{i=1}^N$ and then turns to updating the dictionary. An alternative approach is to update the dictionary after the computation of each representation $\mathbf{x}_i$. Considering again the penalty function posed in (7), the expression $\mathbf{R}\mathbf{d}_S - \mathbf{p}$ stands for its gradient. This gradient accumulates N similar terms, each corresponding to one example $\mathbf{y}_i$. Instead of first computing N sparse representations and then accumulating them into this gradient, we could interlace the two: Assuming that the ith representation has been updated by sparse coding as described in (5), we can update the dictionary by the formula

$$\hat{\mathbf{d}}_S^{new} = \hat{\mathbf{d}}_S^{old} - \mu \left(\sum_{k=1}^{\sqrt{m}} \sum_{l=1}^{\sqrt{m}} x^i_{[k,l]} \mathbf{C}_{[k,l]} \right)^T \left[\sum_{k=1}^{\sqrt{m}} \sum_{l=1}^{\sqrt{m}} x^i_{[k,l]} \mathbf{C}_{[k,l]} \hat{\mathbf{d}}_S^{old} - \mathbf{y}_i \right]. \tag{12}$$

In this update formula, the update term comes from the gradient of the error of the ith example only; i.e., it is the gradient $\mathbf{R}\mathbf{d}_S^{old} - \mathbf{p}$ computed for one example only. The update itself is done in a steepest-descent style. This approach is known by the name *stochastic gradient* (SG). It is practiced quite extensively in adaptive filters (least-mean-squares), computerized tomography, and neural network learning algorithms [5].

The SG approach typically converges faster than the parallel one, and especially in the first several iterations. Each round over all the training data is roughly equivalent in complexity to one update iteration of the previously described algorithm. Furthermore, in scenarios where the input training data includes a drift in its statistics, such an algorithm is able to track these changes over time, and then the step-size parameter μ should be chosen as a small and fixed value. In this work we do not consider such cases but restrict our interest to stationary sources, for which μ should be diminishing as a function of the iteration number, but with a nonintegrable sum [5]. A typical such choice is $\mu = \mu_0/\text{Iteration}$.

Despite its appearance, the update formula in (12) is intuitive and simple to implement. Given the sparse representation $\mathbf{x}_i$ found for the ith example, $\mathbf{y}_i$, we first compute the representation error $\mathbf{e}_i = \mathbf{D}\mathbf{x}_i - \mathbf{y}_i$ (notice that here we chose to return to the representation with the classic dictionary format, with no loss of generality). In fact, it is most likely that the error $\mathbf{e}_i$ is given to us as part of the sparse coding solution. The update formula (12) back-projects this error to the dictionary canvas by subtracting it with the proper coefficients from the existing dictionary $\mathbf{d}_S^{old}$. This is done L times, once per each of the atoms involved in the representation $\mathbf{x}_i$. The term $\mathbf{C}_{[k,l]}^T \mathbf{e}_i$ stands for an image of zeros of size $m \times m$, where the patch $\mathbf{e}^i_{\square}$ is inserted in location $[k,l]$.

The above SG algorithm proposes an update of the ISD after the sparse coding of each example. One could similarly update the ISD after a block of such sparse coding steps. The update formula remains the same, accumulating the derivatives referring to the treated examples. By varying the size of the block, we get that the parallel algorithm and the SG are its two extremes, one for a block size of size N, and the other for blocks having a single example.

The order of the incoming patches influences the convergence rate. When handling a typical image, it is most likely that the lexicographic ordering of patches will be inferior to a random ordering. This is because of the poor variability of patches found in a local

vicinity in the image. Thus, in the experiments reported below we feed the algorithm with a pseudorandom ordering of the image patches.

3.5. Treatment of the mean. In all the above steps we assume that the image patches are treated as given, and no constraints are induced on the dictionary. As already observed in [34] and later practiced in practically every dictionary learning algorithm (e.g., [28, 17, 18, 2, 3]), the mean (DC) of the patches stands as an obstacle to a better representation. Thus, a different treatment of the DC should be practiced.

The DC problem could be easily treated by removing it from the training and testing data and returning it in the image reconstruction. The removal of the DC from the training (and testing examples) is trivial, but this alone is not sufficient, as the training process should take into account the new setting. In reference to (4), one should redefine the operator $\mathbf{C}_{[k,l]}$ to also remove the DC of the extracted patch—this is still a simple linear operator. This will have a direct influence on both the sparse coding stage (where the atoms to be used are DC-free) and the dictionary update formula. All the algorithm implementations (including the comparative studies with the MOD algorithm) presented in this paper use this option.

3.6. Demonstrating the single-scale ISD. We now turn to showing the construction of the ISD for actual images and demonstrating its behavior and properties. We gathered a set of $123^2 = 15,129$ examples, extracted as all the 8×8 patches from the 130×130 pixel image taken from the video sequence Foreman (shown in Figure 2 (left)). An ISD of size 35×35 pixels has been trained for this image using the two proposed algorithms. The parameters chosen in these tests are 10 training iterations, OMP run with $L = 2$, SG run with blocks of 123 examples, and $\mu_0 = 2e - 7$. The two obtained ISDs are shown in Figure 3. Here and elsewhere, the ISDs are visualized by scaling their entries to the range $[0, 1]$ first and then presenting the obtained array as an image.

Figure 2. *The images used to train (left) and test (middle and right). Each patch of size* 8×8 *in the image on the left is an example to use in the training. Similar size patches from the center and right images are used for testing.*

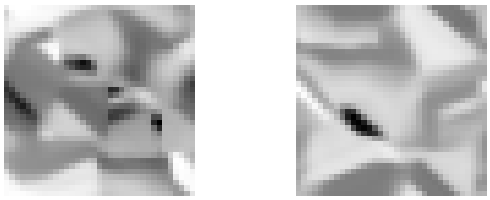

Figure 3. *The two obtained ISDs (left: regular algorithm; right: SG method).*

Beyond the two trained ISDs described above, we have also trained a regular dictionary using the MOD algorithm[3] with a number of atoms chosen as $m = 128$. Note that this dictionary has far fewer atoms than the ISD (128 versus $1,225$). On the other hand, the regular dictionary uses many more parameters, requiring $64 \times 128 = 8,192$, compared to $1,225$ parameters used by the ISD.

Figure 4 presents the representation error[4] as a function of the iteration number, where each iteration corresponds to one full sweep through the examples. The regular (parallel) training algorithms (for the ISD and the MOD) produce two readings of this error per iteration, one after the sparse coding stage and the other after the dictionary update. For the SG algorithm, the representation error is shown after each ISD update (123 times in each sweep). As can be seen, the SG algorithm gives a faster convergence and a better (by 0.5dB) representation error after 10 iterations.

In principle, due to the increased degrees of freedom with the regular dictionary (MOD), one could expect a better representation of the training and testing examples with the MOD results. Nevertheless, we deliberately chose $m = 128$ for the regular dictionary because it leads to roughly the same representation RMSE in the training phase.

In order to see how good the obtained representation is, we study generalization performance, i.e., assessing the average representation error for a new image. Two new sets of $15,129$ patches were extracted from two very different test images taken from the same video sequence (these are shown in Figure 2, center and right[5]).

The representation PSNR for the test images using parallel-trained ISD, the SG approach, and the MOD is shown in Table 1. We provide results for training with 10 and 50 iterations. This table shows that (i) comparable (or even better) representation results are obtained for a new kind of shift-invariance-induced dictionary structure with far fewer degrees of freedom, compared to the direct MOD method; (ii) the SG method is superior in both the training results and the test results; (iii) when operating on a very different image, all methods show a deterioration in representation error, but the worst among these methods is the MOD; and (iv) running the training for more iterations gives slightly better representation errors, and all the above conclusions remain true.

[3]One can choose between the MOD [28, 17] and the K-SVD [2, 3] for this task. Here we used the MOD because it similarly updates the dictionary as a whole in each iteration, leading to a fair comparison.

[4]Presented in peak signal-to-noise ratio (PSNR), obtained by $20\log_{10}(\frac{255}{RMSE})$, where RMSE is the root-mean-squared-error per pixel.

[5]The training frame is the 20th in the sequence, and the test frames are the 40th and the 280th. The last one is chosen to challenge the algorithm with a very different image and see how the error behaves.

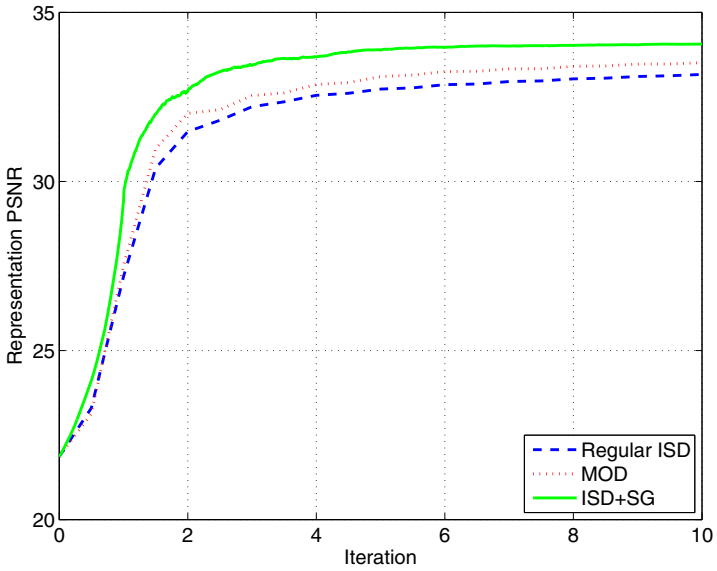

Figure 4. *The training representation PSNR as a function of the iteration.*

Table 1

The representation quality for two test images with the different training models, and for 10 *or* 50 *training iterations.*

	ISD		SG-ISD		MOD	
	10 iter.	50 iter.	10 iter.	50 iter.	10 iter.	50 iter.
Test 1	33.38 dB	33.71 dB	33.88 dB	33.99 dB	33.09 dB	33.29 dB
Test 2	29.61 dB	29.72 dB	29.75 dB	29.73 dB	29.08 dB	29.09 dB

3.7. Multiscale ISD training and usage. An important benefit of the ISD structure is its ability to handle signals of varying sizes by extracting flexible size atoms from the signature. This multiscale flexibility cannot be matched by the regular dictionary concept, where different size atoms require training and storage of several separate dictionaries. In the following we will follow closely the formulation shown in section 3.1 and extend the learning algorithm to varying scales. While the following discussion is presented for the completeness of the presentation, in this paper we do not explore experimentally the multiscale option of the ISD. A separate work is required for studying this option and the potential it brings.

We assume that we are given a set of training patches of varying sizes, $\{\mathbf{y}_{\square}^{i}\}_{i=1}^{N} \in \mathbb{R}^{\sqrt{n_i}\times\sqrt{n_i}}$ (or their vector representations $\{\mathbf{y}_i\}_{i=1}^{N}$). We aim to find an ISD of specific size $\mathbf{D}_S \in \mathbb{R}^{\sqrt{m}\times\sqrt{m}}$ that leads to the smallest representation error for each patch, while using L_i (may vary!)

atoms. Similarly to the problem defined in (4), we define an energy function of the form

$$(13)\quad \hat{\mathbf{d}}_S = \operatorname{Arg}\min_{\mathbf{d}_S} \sum_{i=1}^{N} \epsilon_i^2(\mathbf{d}_S) \quad \text{subject to}$$

$$\epsilon_i^2(\mathbf{d}_S) = \min_{\mathbf{x}} \left\| \mathbf{y}_i - \sum_{k=1}^{\sqrt{m}} \sum_{l=1}^{\sqrt{m}} x_{[k,l]} \mathbf{C}_{[k,l]}^{n_i} \mathbf{d}_S \right\|_2^2 \quad \text{subject to} \quad \|\mathbf{x}\|_0 \le L_i,\ 1 \le i \le N.$$

Compared with (4), this formulation treats input patches $\mathbf{y}_i$ of varying sizes, and with possibly varying number of allocated atoms. Consequently, the operator $\mathbf{C}_{[k,l]}^{n_i}$ extracts a patch of size $\sqrt{n_i} \times \sqrt{n_i}$ from location $[k,l]$ to become an atom in the composed representation.

Due to the close resemblance to previous formulation, there is no point in repeating all of the description of the training algorithm in both the parallel and the SG modes—they are almost exactly the same, with small modifications to account for different size patches. Note that the sparse coding stage is done on each example separately, and thus there is no mixture of varying size atoms that might complicate this step. The dictionary update remains the same, with the operators $\mathbf{C}_{[k,l]}^{n_i}$ replacing the fixed-sized operators.

A problem that remains unsolved at this stage is the question of how many examples to use from each size and how to allocate L_i to each accordingly. We do not propose in this paper a solution for this problem, as it depends on the application in use.

Another possibility that we do not explore in this work is a true mixture of scales in the composition of signals: If we aim to represent an image of size $\sqrt{n_0} \times \sqrt{n_0}$ pixels, we could, in principle, compose it with atoms of varying sizes that are extracted from the ISD. When an atom of size $\sqrt{n_1} \times \sqrt{n_1}$ is used, where $n_1 \le n_0$, this implies that this atom covers a portion of the image in a specific location.

4. Image denoising with ISD.

We test the applicability of the ISD by experimenting on the image denoising problem. Generally speaking, we adopt a method similar to the one suggested in [15, 16], which led to state-of-the-art denoising results using a "regular" dictionary. In the following experiments we restrict our algorithms to the single-scale ISD, with the understanding that better performance *might* be obtained when exploiting the multiscale freedom that ISD provides.

Let $\tilde{\mathbf{I}}$ be a corrupted version of the image $\mathbf{I}$, after the addition of white zero-mean Gaussian noise with power σ_n. The first algorithm we consider is built of the following steps:

- *Training.* A global ISD is trained on patches of a set of appropriate images that do not belong to the test set. We used an ISD of size 75×75 pixels, trained on single-scale patches of size 8×8. This ISD is presented in Figure 5. It was obtained by training on $110,000$ examples of 8×8 pixels, extracted from a set of 10 arbitrary images, using 10 iterations and with $L = 2$ (number of atoms per example in the training set).
- *Sparse coding.* Each patch $\mathbf{y}_i$ of size 8×8 from the noisy image is extracted and sparse-coded using OMP and the obtained ISD. We consider all the possible patches with overlaps. The OMP accumulates atoms until a representation error of $C \cdot \sigma_n$ is reached ($C = 1.1$). This leads to an approximate patch with reduced noise, $\hat{\mathbf{x}}_i$.

Figure 5. *ISD of size* 75×75*, trained on patches of size* 8×8 *from several images.*

- *Averaging.* The output image is obtained by adding the patches $\hat{\mathbf{x}}_i$ in their proper locations and averaging the contributions in each pixel (a more sophisticated averaging technique that can be applied here is suggested in [25, 10]).

We refer to this algorithm as the *global* ISD, as it uses a single ISD to fit all images.

The obtained ISD requires only a third of the space required by the global dictionary in [16], and this has a direct and crucial impact on the complexity of the dictionary update stage in the MOD algorithm [28, 17, 18] versus the ISD algorithm, since this number reflects the size of the matrix to be inverted.[6] Indeed, this gap in memory and computational requirements is further expanded when considering the SG algorithm.

A better denoising scheme, introduced originally in [16], is one that trains the ISD using the overlapping patches from the noisy image directly. While the training data in such a setting is noisy, the potential of using a dictionary that is well fitted to the image content seems to be promising. As shown in [16], this, in fact, leads to state-of-the-art denoising performances, competing favorably with the results reported in [36]. A possible explanation for the success of such a scheme may be the fact that while the clean image exhibits a repetitive and correlated structure, the noise does not enjoy such a property. The ISD learning has a cleaning feature built into it, due to the many averaged patches formed in (9) and (10).

We adopted this idea, this time with the ISD and using the same parameters as above (block size 8×8 pixels, $L = 2$, 10 iterations). The results of this scheme are summarized in Table 2 for a set of six test images and a wide range of noise powers—this test suite is the one proposed in [36] and later followed by a group of papers (including [16]). Two of the test images are shown in Figure 6. This table also shows the results of two alternative algorithms:

- A regular dictionary-based algorithm that employs the MOD dictionary learning (10 iterations, as before). This comparison comes to test the role of shift-invariance in the dictionary structure in the denoising performance. The MOD algorithm is deployed with 90 atoms, so as to get a comparable number of free parameters in the dictionary ($8 \times 8 \times 90$ is comparable to the 75^2 pixels in the ISD). Note that for this choice of

[6] As we show below, the comparative experiments we provide with MOD use the same number of free parameters in the dictionaries, so as to resolve this gap.

Table 2

The denoising results in dB for six test images and a noise power in the range $[5, 100]$ *gray values. For each test setting, three results are provided: the ISD algorithm results (top); the MOD comparable algorithm, using regular dictionary and using the same degrees of freedom (middle); and the epitome denoising method (bottom). The best among each three results is highlighted. The rightmost column gives the average gain in dB from the noisy image to the denoised one for the six test images.*

$\sigma/PSNR$	Lena	Barb	Boats	Fgrpt	House	Peppers	Aver. Diff
5/34.15	**38.41**	37.76	37.00	**36.61**	**39.42**	**37.76**	**3.68**
	38.39	**37.92**	**37.11**	36.60	39.19	37.43	3.62
	29.99	27.01	27.56	26.30	31.13	25.86	-4.51
10/28.13	**35.42**	**34.21**	**33.64**	**32.46**	**36.05**	**34.23**	**6.21**
	35.17	34.00	33.49	32.44	35.63	33.92	5.98
	30.35	27.28	27.97	26.33	31.57	26.98	0.28
15/24.61	**33.64**	**32.22**	**31.79**	**30.16**	**34.25**	**32.30**	**7.77**
	33.23	31.71	31.50	30.11	33.43	31.76	7.35
	30.37	27.38	27.99	26.30	31.56	27.31	3.88
20/22.11	**32.25**	**30.71**	**30.41**	**28.56**	**32.72**	**30.69**	**8.78**
	31.73	30.20	29.99	28.44	32.01	30.20	8.32
	29.96	27.23	27.71	26.14	31.06	26.81	6.04
25/20.17	**31.09**	**29.22**	28.45	27.15	**31.76**	29.44	**9.34**
	30.64	28.94	**28.91**	**27.20**	30.77	**29.03**	9.08
	29.46	27.00	27.39	25.83	30.06	26.71	7.57
50/14.15	**27.31**	25.17	**25.77**	23.34	26.26	25.03	**11.33**
	26.67	25.10	25.34	23.56	**26.76**	**25.29**	11.30
	24.55	**26.60**	25.20	**23.79**	25.77	23.91	10.82
75/10.63	**24.85**	**22.50**	**23.34**	20.21	23.02	22.61	12.13
	24.23	22.81	23.17	21.07	**24.10**	**22.88**	**12.42**
	24.15	22.40	23.18	**21.60**	22.34	21.19	11.85
100/8.13	**22.90**	21.00	**21.80**	18.39	20.54	19.85	12.62
	22.42	20.97	21.46	19.28	**21.84**	**20.88**	**13.01**
	20.88	**22.29**	21.41	**19.90**	20.09	19.03	12.47

Figure 6. *Two of the six test images we used for the various denoising tests (Lena (left) and Barbara (right)).*

number of atoms in the MOD, the computational complexity of the ISD algorithm is lower, due to algorithmic shortcuts explained below.

- The epitome-based denoising algorithm, as described in [26, 8]. We used an existing code taken from http://www.psi.toronto.edu/~vincent/sourcecode.html.

Two kinds of results are deliberately omitted from Table 2 in order not to clutter it. The results of the global ISD denoising are not included as they are inferior to the adaptive ones by 0–2.5dB (depending on the noise power—the gap is larger for weaker noise). Also, the results reported in [16, 15] are not included, since those roughly parallel the MOD results shown here, with ±0.5dB fluctuations.

As can be seen from Table 2, the results of the (adaptive) ISD are the best for most images and noise powers. An exception to this rule takes place for very high noise power where the MOD seems to perform slightly better. The epitome denoising is far behind for low noise power and gets competitive results for very strong noise. We should add that we expected the ISD (and the MOD) to show an improvement in their performance in high noise powers by reducing the number of free parameters, but this was not verified experimentally.

When running the MOD and the ISD denoising algorithms with 50 iterations, in an attempt to get converged dictionaries, the results did not change by much. We observed a small improvement ($\approx$ 0.1dB on average) for all algorithms, which does not change the above conclusions and definitely does not justify the added complexity.

Figure 7 presents the ISDs and the epitomes obtained for the images Lena and Barbara shown in Figure 6. As can be seen, both the ISDs and the epitomes are very different from one image to the other, and are well adapted to the type of image they serve. Figure 8 presents a selected set of denoising results to illustrate the differences between the various algorithms.

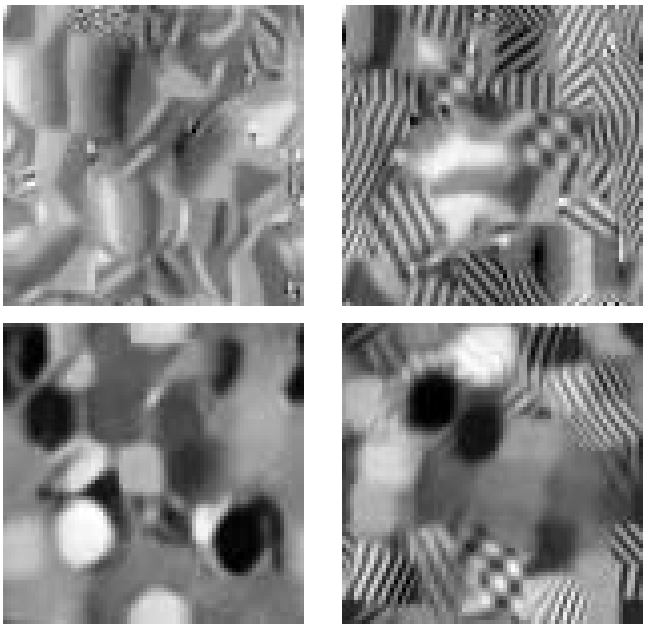

Figure 7. *Trained ISDs (top row) and epitomes (bottom row) for two images, Lena (left) and Barbara (right). These were obtained while denoising these images with* $\sigma = 15$.

Figure 8. *Selected denoising results. These results refer to noise power $\sigma = 15$, showing the noisy image, the ISD, the MOD, and the epitome denoising results.*

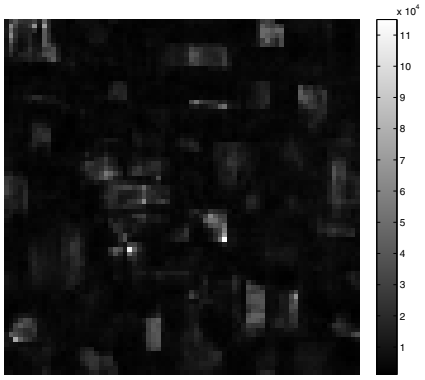

Figure 9. *A map of the relative usage of atoms in the ISD for Barbara for the denoising experiment with noise power* $\sigma = 15$.

How are the different atoms in the ISD used in the image formation? In order to answer this question and validate that all the atoms are indeed used, we performed the following: For each pixel in the ISD we accumulated the absolute values of the coefficients referring to the patch centered around it. The resulting usage image for Barbara is presented in Figure 9, giving a ratio of 93.7 between the maximum and minimum. This shows that while there is a variance in the usage of different atoms in the ISD, all are effectively used.

We return now to the computational complexity issue. The most time-consuming part in the training process of both a regular dictionary and an ISD is the sparse representation part, done with OMP. In the tested application, the representation error is reduced as more atoms are added, until a certain representation error is reached ($C \cdot \sigma_n$). Using an ISD, we can exploit the fact that the support of the previously represented neighbor patch is known and that shifted versions of all its atoms can be extracted with no cost. When representing a patch $\mathbf{C}_{[k,l]}\mathbf{I}_S$, where $\mathbf{I}_S$ is an arrangement of the image $\mathbf{I}$ as a column, we start by exploring the set of atoms that represented its neighbor patch $\mathbf{C}_{[k-1,l]}\mathbf{I}_S$ (or $\mathbf{C}_{[k,l-1]}\mathbf{I}_S$—when representing patches sequentially, one of these neighbors, or even both, should have already been represented). Denote the atoms involved in the construction of the neighbor patch as $\mathbf{d}_1, \mathbf{d}_2, \ldots, \mathbf{d}_L$. Then each of these atoms can be shifted inside the ISD one pixel down or right (depending on the location of the neighbor patch), resulting in a new set of atoms likely to represent well the current patch. We then project the current patch onto the potential set of atoms and check the representation error. If it is smaller than $C \cdot \sigma_n$, we move onto the next patch without applying the OMP algorithm. Otherwise, we calculate the representation from start using OMP. This variation of the algorithm can be applied only when working with the ISD, and it results in a substantial savings in computations.

Figure 10 presents the number of calls to the OMP algorithm in each iteration of the training process on the image House. As can be seen, the larger the noise level, the fewer calls are required. Also, as the dictionary is better trained, fewer calls are required as well. Notice that the number of patches in this image is $(256 - 8 + 1)^2 = 62,001$, whereas the number of required OMP processes in each iteration is between 3%–40% of this number (depending on the noise level and iteration number). In the last iteration we represent all patches from start

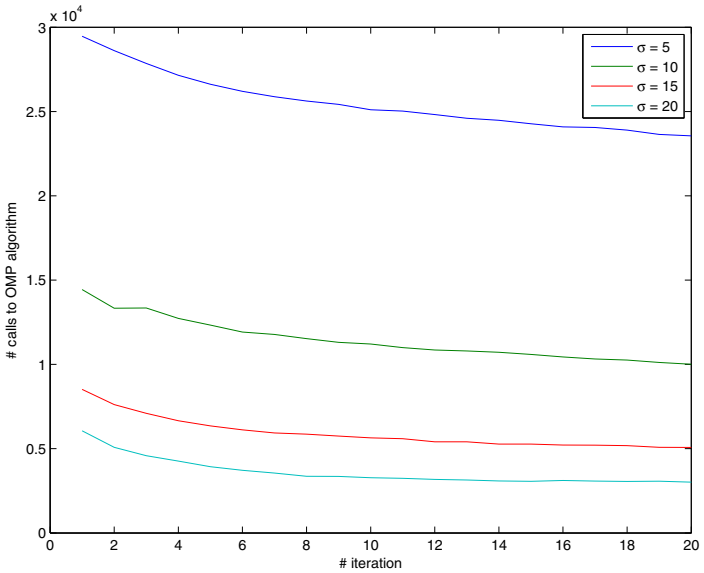

Figure 10. *The number of calls to the OMP algorithm as a function of the training iterations on the image House.*

to finalize the denoising process.

The bottom line comparison between the MOD- and the ISD-based denoising in the above settings is the following:

- As the two methods use the same number of free parameters in their dictionaries, the dictionary update stage of the two is of the same complexity.
- If we choose to employ the SG update for the ISD, a gain factor of m can be obtained (m is the number of pixels in the ISD).
- The OMP for the both algorithms requires $\mathcal{O}(mL)$ computations (L is the number of atoms in the obtained representation).
- Due to the shortcut described above based on the shift-invariance property, the ISD gains speed by a factor of 2–10, due to the reduced number of OMP computations required.

Assuming that the dictionary update stage is negligible, the bottom line outcome is that the ISD algorithm is 2–10 times faster than the MOD method for the case of equivalent number of free parameters. As we have seen, this setting also leads to better denoising performance by the ISD.

5. Conclusion. In this paper we have described the Image Signature Dictionary, a structure of a dictionary for sparse and redundant representations that extends the epitomes [26, 8]. We have shown how such a dictionary can be trained and used, and demonstrated several advantages it provides over the use of a "regular" dictionary and over epitomes. We believe that the important features of the ISD make it a promising alternative to the classical dictionary structure. Further investigation is needed to see how to exploit its scale flexibility.

REFERENCES

[1] M. Aharon, *Overcomplete Dictionaries for Sparse Representation of Signals*, Ph.D. thesis, The Technion–Israel Institute of Technology, Haifa, Israel, 2006.

[2] M. Aharon, M. Elad, and A.M. Bruckstein, *K-SVD: An algorithm for designing of overcomplete dictionaries for sparse representation*, IEEE Trans. Signal Process., 54 (2006), pp. 4311–4322.

[3] M. Aharon, M. Elad, and A.M. Bruckstein, *On the uniqueness of overcomplete dictionaries, and a practical way to retrieve them*, Linear Algebra Appl., 416 (2006), pp. 48–67.

[4] J.F. Bard, *Practical Bilevel Optimization*, Kluwer Academic Publishers, Dordrecht, The Netherlands, 1998.

[5] D.P. Bertsekas, *Nonlinear Programming*, Athena Scientific, Belmont, MA, 1995.

[6] T. Blumensath and M. Davies, *Sparse and shift-invariant representations of music*, IEEE Trans. Audio, Speech, and Language Processing, 14 (2006), pp. 50–57.

[7] S.S. Chen, D.L. Donoho, and M.A. Saunders, *Atomic decomposition by basis pursuit*, SIAM J. Sci. Comput., 20 (1998), pp. 33–61.

[8] V. Cheung, B.J. Frey, and N. Jojic, *Video epitomes*, in Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), 2005, pp. 42–49.

[9] R.R. Coifman and M.V. Wickerhauser, *Adapted waveform analysis as a tool for modeling, feature extraction, and denoising*, Optical Engineering, 33 (1994), pp. 2170–2174.

[10] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian, *Image denoising by sparse* 3*D transform-domain collaborative filtering*, IEEE Trans. Image Process., 16 (2007), pp. 2080–2095.

[11] M.N. Do and M. Vetterli, *The contourlet transform: An efficient directional multiresolution image representation*, IEEE Trans. Image Process., 14 (2005), pp. 2091–2106.

[12] D.L. Donoho and M. Elad, *Optimally sparse representation in general (nonorthogonal) dictionaries via* ℓ^1 *minimization*, Proc. Nat. Acad. Sci. USA, 100 (2003), pp. 2197–2202.

[13] D.L. Donoho and M. Elad, *On the stability of the basis pursuit in the presence of noise*, Signal Process., 86 (2006), pp. 511–532.

[14] D.L. Donoho, M. Elad, and V. Temlyakov, *Stable recovery of sparse overcomplete representations in the presence of noise*, IEEE Trans. Inform. Theory, 52 (2006), pp. 6–18.

[15] M. Elad and M. Aharon, *Image denoising via learned dictionaries and sparse representation*, in Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), New York, 2006.

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

[17] K. Engan, S.O. Aase, and J.H. Husøy, *Method of optimal directions for frame design*, in Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing, Vol. 5, 1999, pp. 2443–2446.

[18] K. Engan, K. Skretting, and J.H. Husøy, *Family of iterative LS-based dictionary learning algorithms, ILS-DLA, for sparse signal representation*, Digit. Signal Process., 17 (2007), pp. 32–49.

[19] R.M. Figueras, I. Ventura, O. Divorra-Escoda, and P. Vandergheynst, *A Matching Pursuit Full Search Algorithm for Image Approximations*, ITS Technical Report TR-ITS-31/2004, Signal Processing Institute, Lausanne, Switzerland, 2004.

[20] J.J. Fuchs, *On sparse representations in arbitrary redundant bases*, IEEE Trans. Inform. Theory, 50 (2004), pp. 1341–1344.

[21] J.J. Fuchs, *Recovery of exact sparse representations in the presence of bounded noise*, IEEE Trans. Inform. Theory, 51 (2005), pp. 3601–3608.

[22] I.F. Gorodnitsky and B.D. Rao, *Sparse signal reconstruction from limited data using FOCUSS: A re-weighted norm minimization algorithm*, IEEE Trans. Signal Process., 45 (1997), pp. 600–616.

[23] R. Gribonval, R. Figueras, and P. Vandergheynst, *A simple test to check the optimality of a sparse signal approximation*, Signal Process., 86 (2006), pp. 496–510.

[24] R. Gribonval and M. Nielsen, *Sparse decompositions in unions of bases*, IEEE Trans. Inform. Theory, 49 (2003), pp. 3320–3325.

[25] O.G. Guleryuz, *Weighted averaging for denoising with overcomplete dictionaries*, IEEE Trans. Image Process., 16 (2007), pp. 3020–3034.

[26] N. Jojic, B.J. Frey, and A. Kannan, *Epitomic analysis of appearance and shape*, in Proceedings of the IEEE International Conference on Computer Vision (ICCV), 2003, pp. 34–41.
[27] A. Kannan, J. Winn, and C. Rother, *Clustering appearance and shape by learning jigsaws*, in Advances in Neural Information Processing Systems (NIPS), 2006, pp. 657–664.
[28] K. Kreutz-Delgado, J.F. Murray, B.D. Rao, K. Engan, T.-W. Lee, and T.J. Sejnowski, *Dictionary learning algorithms for sparse representation*, Neural Comput., 15 (2003), pp. 349–396.
[29] J. Lasserre, A. Kannan, and J. Winn, *Hybrid learning of large jigsaws*, in Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), 2007, pp. 1–8.
[30] J. Mairal, M. Elad, and G. Sapiro, *Sparse representation for color image restoration*, IEEE Trans. Image Process., 17 (2008), pp. 53–69.
[31] S. Mallat and E. LePennec, *Sparse geometric image representation with bandelets*, IEEE Trans. Image Process., 14 (2005), pp. 423–438.
[32] S. Mallat and Z. Zhang, *Matching pursuits with time-frequency dictionaries*, IEEE Trans. Signal Process., 41 (1993), pp. 3397–3415.
[33] B.K. Natarajan, *Sparse approximate solutions to linear systems*, SIAM J. Comput., 24 (1995), pp. 227–234.
[34] B.A. Olshausen and B.J. Field, *Emergence of simple-cell receptive field properties by learning a sparse code for natural images*, Nature, 381 (1997), pp. 607–609.
[35] Y.C. Pati, R. Rezaiifar, and P.S. Krishnaprasad, *Orthogonal matching pursuit: Recursive function approximation with applications to wavelet decomposition*, in Proceedings of the 27th Annual Asilomar Conference on Signals, Systems, and Computers, 1993, pp. 40–44.
[36] J. Portilla, V. Strela, M.J. Wainwright, and E.P. Simoncelli, *Image denoising using scale mixtures of Gaussians in the wavelet domain*, IEEE Trans. Image Process., 12 (2003), pp. 1338–1351.
[37] J.G. Proakis and D.K. Manolakis, *Digital Signal Processing: Principles, Algorithms and Applications*, 3rd ed., Prentice-Hall, Englewood Cliffs, NJ, 1995.
[38] M. Protter and M. Elad, *Image sequence denoising via sparse and redundant representations*, IEEE Trans. Image Process., to appear.
[39] K. Skretting and J.H. Husøy, *Texture classification using sparse frame-based representation*, EURASIP J. Appl. Signal Process., 2006 (2006), article ID 52561.
[40] J.L. Starck, E.J. Candès, and D.L. Donoho, *The curvelet transform for image denoising*, IEEE Trans. Image Process., 11 (2002), pp. 670–684.
[41] J.A. Tropp, *Greed is good: Algorithmic results for sparse approximation*, IEEE Trans. Inform. Theory, 50 (2004), pp. 2231–2242.
[42] J.A. Tropp, *Just relax: Convex programming methods for subset selection and sparse approximation*, IEEE Trans. Inform. Theory, 52 (2006), pp. 1030–1051.