

Sparse Representation for Color Image Restoration

Julien Mairal, *Member, IEEE*, Michael Elad, *Member, IEEE*, and Guillermo Sapiro, *Senior Member, IEEE*

Abstract—Sparse representations of signals have drawn considerable interest in recent years. The assumption that natural signals, such as images, admit a sparse decomposition over a redundant dictionary leads to efficient algorithms for handling such sources of data. In particular, the design of well adapted dictionaries for images has been a major challenge. The K-SVD has been recently proposed for this task [1] and shown to perform very well for various grayscale image processing tasks. In this paper, we address the problem of learning dictionaries for color images and extend the K-SVD-based grayscale image denoising algorithm that appears in [2]. This work puts forward ways for handling nonhomogeneous noise and missing information, paving the way to state-of-the-art results in applications such as color image denoising, demosaicing, and inpainting, as demonstrated in this paper.

Index Terms—Color processing, denoising, demosaicing, image decomposition, image processing, image representations, inpainting, sparse representation.

I. INTRODUCTION

IN signal and image processing, we often impose an arbitrary model to describe the data source. Such a model becomes paramount when developing algorithms for processing these signals. In this context, Markov random field (MRF), principal component analysis (PCA), and other related techniques are popular and often used.

The *Sparseland* model is an emerging and powerful method to describe signals based on the sparsity and redundancy of their representations [2], [3]. For signals from a class $\Gamma \subset \mathbb{R}^n$, this model suggests the existence of a specific dictionary (i.e., a matrix) $\mathbf{D} \in \mathbb{R}^{n \times k}$ which contains k prototype signals, also referred to as atoms. The model assumes that for any signal $\mathbf{x} \in \Gamma$, there exists a sparse linear combination of atoms from \mathbf{D} that approximates it well. Put more formally, this reads $\forall \mathbf{x} \in \Gamma, \exists \alpha \in \mathbb{R}^k$ such that $\mathbf{x} \approx \mathbf{D}\alpha$ and $\|\alpha\|_0 \ll n$.

The notation $\|\cdot\|_0$ is the ℓ^0 -quasi-norm, which counts the number of nonzero elements in a vector. We typically assume that $k > n$, implying that the dictionary \mathbf{D} is redundant in describing \mathbf{x} .

If we consider the case where Γ is the set of natural images, dictionaries such as wavelets of various sorts [4], curvelets [5],

[6], contourlets [7], [8], wedgelets [9], bandlets [10], [11], and steerable wavelets [12], [13], are all attempts to design dictionaries that fulfill the above model assumption. Indeed, these various transforms have led to highly effective algorithms in many applications in image processing, such as compression [14], denoising [15]–[18], inpainting [19], and more. Common to all these predefined dictionaries is their analytical nature, and their reliance on the geometrical nature of natural images, especially piece-wise smooth ones.

In [1], the authors introduce the K-SVD algorithm, a way to learn a dictionary, instead of exploiting predefined ones as described above, that leads to sparse representations on training signals drawn from Γ . This algorithm uses either orthogonal matching pursuit (OMP) [20]–[22], or basis pursuit (BP), [23], as part of its iterative procedure for learning the dictionary. The follow-up work reported in [2] and [3] proposes a novel and highly effective image denoising algorithm for the removal of additive white Gaussian noise with grayscale images. Their proposed method includes the use of the K-SVD for learning the dictionary from the noisy image directly.

Greedy algorithms like matching pursuit (MP) and its variations are commonly used to retrieve sparse approximations of multichannel signals, with fixed dictionaries and standard Euclidean metric, for various other modalities and applications, e.g., for compression of color images [24] and for audio source separation [25], [26].

In this paper, our main aim is to extend the algorithm reported in [2] to color images (and to vector-valued images, in general), and then show the applicability of this extension to other inverse problems in color image processing. The extension to color can be easily performed by a simple concatenation of the RGB values to a single vector and training on those directly, which gives already better results than denoising each channel separately. However, such a process produces false colors and artifacts, which are typically encountered in color image processing. The first part of this work presents a method to overcome these artifacts, by adapting the OMP inner-product (metric) definition.

This paper also describes an extension of the denoising algorithm to the proper handling of nonhomogeneous noise. This development is crucial in cases of missing values, such as in the color image demosaicing and the inpainting problems. Treating the missing values as corrupted by a strong (impulse) noise, our general setting fits both problems. We demonstrate the success of the proposed scheme in demosaicing and inpainting, as well as color denoising applications, exhibiting in all cases state-of-the-art results. We also show that interacting between the learning and the restoration further improves the color image processing results.

This paper is organized as follows. In Section II, we describe prior art on example-based image denoising (and general image

Manuscript received October 23, 2006; revised October 8, 2007. This work was supported in part by the National Science Foundation, The Office of Naval Research, NGA, DARPA, the McKnight Foundation, and the Israel Science Foundation under Grant 796/05. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Gabriel Marcu.

J. Mairal and G. Sapiro are with the Department of Electrical and Computer Engineering, University of Minnesota, Minneapolis, MN 55455 USA (e-mail: julien.mairal@m4x.org; guille@ece.umn.edu).

M. Elad is with the Department of Computer Science, The Technion—Israel Institute of Technology, Haifa 32000, Israel (e-mail: elad@cs.technion.ac.il).

Color versions of one or more of the figures in this paper are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/TIP.2007.911828

processing) methods. Section III is devoted to a brief description of the K-SVD-based grayscale image denoising algorithm as proposed in [2]. Section IV describes the novelties offered in this paper—the extension to color images, the handling of color artifacts, and, finally, the treatment of nonhomogeneous noise, along with its relation to demosaicing and inpainting. In Section V, we provide various experiments that demonstrate the effectiveness of the proposed algorithms. Section VI concludes this paper with a brief description of its contributions and a list of open questions for future work, including preliminary discussion and results on how to extend the above to a multiscale algorithm.

II. PRIOR ART

Many problems in image processing and computer vision are in a dire need for prior models of the images they handle. This is especially true whenever information is missing, damaged, or modified. Armed with a good generic image prior, restoration algorithms become very effective. The research activity on the topic of image priors is too wide to be compacted into this paper, and as our interest is primarily in learned priors, such general survey is beyond its scope anyhow. We, therefore, restrict our discussion to recent methods that lean on image examples in their construction of the prior.

When basing the learned prior on image examples, the first junction to cross is the one which splits between parametric and nonparametric prior models. The parametric path suggests an analytical expression for the prior, and directs the learning process to tune the prior parameters based on examples. Such is the case with the MRF prior learned in [27] and later in [28] and [29]; the wavelet based image prior as appears in [30]; the Tikhonov regularization proposed by Haber and Tenorio [31]; the super-resolution approach adopted by Baker and Kanade [32]; and the recent K-SVD denoising as described in [2] and [3].

The alternative path, a nonparametric learning, uses image examples directly within the reconstruction process, as practiced by Efros and Leung for texture synthesis [33], by Freeman *et al.* for super-resolution [34], [35], and by several follow-up works [36]–[39] for super-resolution and inpainting. Interestingly, most of the above direct methods avoid the prior and target instead the posterior density, from which reconstruction is easily obtained.

The second major junction to cross in exploiting examples refers to the question of the origin of these examples. Many of the above-described works use a separate corpus of training images for learning the prior (or its parameters). The alternative option is to use examples from the corrupted image itself. This surprising idea has been proposed in [40] as a universal denoiser of images, which learns the posterior from the given image in a way inspired by the Lempel–Ziv universal compression algorithm. Another path of such works is the nonlocal means [41], [42] and related works [43], [44]. Interestingly, the work in [2] belongs to this family as well, as the dictionary can be based on the noisy image itself.

Most of the above methods deploy processing of small image patches, a theme that characterizes our method as well. In this paper, we present a framework for learning a sparsifying dic-

tionary for color image patches taken from natural images. As such, the approach we adopt is the parametric one. The learning we propose leans on both external data-set, as well as on the damaged image directly. The novelty of this paper is in the way of introducing the color to the K-SVD algorithm such that it avoids typical artifacts, and its extension to nonhomogeneous noise, which enables the handling of color inpainting and demosaicing.

The only work we are aware of, which handles color images within the framework of parametric learned models, is the one reported in [28]. This work builds on the *field of experts*, as developed in [29], to learn an MRF model for color images, and uses it successfully for image denoising. The main theme of this paper is an attempt to circumvent the high dimensionality of the training space by several simplifications over the original method in [29]. Color artifacts are not mentioned, and, indeed, the shown results demonstrate a phenomenon we have experienced, too, of a tendency to washout the color content of the image. We address this in this paper by proposing a new metric in the sparsity representation. We will return to this work in the experimental results section, and show comparisons of performance favorable to our framework.

III. GRAYSCALE K-SVD DENOISING ALGORITHM

In [2] and [3], Aharon and Elad present a K-SVD-based algorithm for denoising of grayscale images with additive homogeneous white Gaussian noise. We now briefly review the main mathematical framework of this approach, as our work builds upon it. First, let \mathbf{x}_0 be a clean image written as a column vector of length N . Then one considers its noisy version

$$\mathbf{y} = \mathbf{x}_0 + \mathbf{w}$$

where \mathbf{w} is a white Gaussian noise with a spatially uniform deviation σ , which is assumed to be known. Given fixed-size patches $\sqrt{n} \times \sqrt{n}$, one assumes that all such patches in the clean image \mathbf{x}_0 admit a sparse representation. Addressing the denoising problem as a sparse decomposition technique per each patch leads to the following energy minimization problem:

$$\{\hat{\alpha}_{ij}, \hat{\mathbf{D}}, \hat{\mathbf{x}}\} = \arg \min_{\mathbf{D}, \alpha_{ij}, \mathbf{x}} \lambda \|\mathbf{x} - \mathbf{y}\|_2^2 + \sum_{i,j} \mu_{ij} \|\alpha_{ij}\|_0 + \sum_{i,j} \|\mathbf{D}\alpha_{ij} - \mathbf{R}_{ij}\mathbf{x}\|_2^2. \quad (1)$$

In this equation, $\hat{\mathbf{x}}$ is the estimator of \mathbf{x}_0 , and the dictionary $\hat{\mathbf{D}} \in \mathbb{R}^{n \times k}$ is an estimator of the optimal dictionary which leads to the sparsest representation of the patches in the recovered image. The indices $[i, j]$ mark the location of the patch in the image (representing its top-left corner). The vectors $\hat{\alpha}_{ij} \in \mathbb{R}^k$ are the sparse representations for the $[i, j]$ th patch in $\hat{\mathbf{x}}$ using the dictionary $\hat{\mathbf{D}}$. The operator \mathbf{R}_{ij} is a binary $n \times N$ matrix which extracts the square $\sqrt{n} \times \sqrt{n}$ patch of coordinates $[i, j]$ from the image written as a column vector of size N .

The first term in (1) introduces the likelihood force that demands a proximity between \mathbf{x} and \mathbf{y} . The second and the third terms together pose the image prior. This regularization term assumes that good-behaved natural images are to exhibit a sparse representation for every patch, and from every location in the

Parameters: λ (Lagrange multiplier); C (noise gain); J (the number of iterations); k (size of the dictionary); n (size of the patches and the atoms).

Initialization: Set $\hat{\mathbf{x}} = \mathbf{y}$; Let $\hat{\mathbf{D}} = (\hat{\mathbf{d}}_l)_{l \in 1 \dots k}$ be some initial dictionary.

Loop: Repeat J times

- *Sparse Coding:* Fix $\hat{\mathbf{D}}$ and use OMP to compute $\hat{\alpha}_{ij}$ per each patch by solving

$$\forall ij \quad \hat{\alpha}_{ij} = \arg \min_{\alpha_{ij}} \|\alpha_{ij}\|_0 \text{ subject to } \|\mathbf{R}_{ij}\hat{\mathbf{x}} - \hat{\mathbf{D}}\alpha_{ij}\|_2^2 \leq n(C\sigma)^2. \quad (2)$$

- *Dictionary Update:* Fix all $\hat{\alpha}_{ij}$, and for each atom $l \in 1, 2, \dots, k$ in $\hat{\mathbf{D}}$,
 - Select the set of patches which use this atom, $\omega_l = \{[i, j] | \hat{\alpha}_{ij}(l) \neq 0\}$.
 - For each patch $[i, j] \in \omega_l$, compute its residual, $e_{ij}^l = \mathbf{R}_{ij}\hat{\mathbf{x}} - \hat{\mathbf{D}}\hat{\alpha}_{ij} + \hat{\mathbf{d}}_l\hat{\alpha}_{ij}(l)$.
 - Set \mathbf{E}_l as the matrix whose columns are the \mathbf{e}_{ij}^l , and $\hat{\alpha}^l$ the row vector whose elements are the $\hat{\alpha}_{ij}(l)$.
 - Update $\hat{\mathbf{d}}_l$ and the $\hat{\alpha}_{ij}(l)$ by minimizing:

$$(\hat{\mathbf{d}}_l, \hat{\alpha}^l) = \arg \min_{\alpha^l, \|\mathbf{d}_l\|_2=1} \|\mathbf{E}_l - \mathbf{d}_l\alpha^l\|_F^2. \quad (3)$$

This rank-one approximation is performed by a truncated Singular Value Decomposition (SVD) of the matrix \mathbf{E}_l .

Averaging: Perform a weighted average:

$$\hat{\mathbf{x}} = \left(\lambda \mathbf{I} + \sum_{ij} \mathbf{R}_{ij}^T \mathbf{R}_{ij} \right)^{-1} \left(\lambda \mathbf{y} + \sum_{ij} \mathbf{R}_{ij}^T \hat{\mathbf{D}} \hat{\alpha}_{ij} \right). \quad (4)$$

Fig. 1. K-SVD-based image denoising algorithm as proposed in [2].

image, over the learned dictionary $\hat{\mathbf{D}}$. The second term provides the sparsest representation, and the third term ensures the consistency of the decomposition. The choice of the norm \mathbb{L}^2 is relatively arbitrary and could be changed in this formulation, as later proposed in this paper for color images. Note that norms different than \mathbb{L}^2 might lead to difficulties in the minimization.

To approximate a solution for this complex minimization task, the authors of [2] and [3] put forward an iterative method that incorporates the K-SVD algorithm, as presented in [1]. Fig. 1 presents this image denoising algorithm in details.

One can notice that different steps in this algorithm reject the noise. First, the OMP stops when the approximation reaches a sphere of radius $\sqrt{n}C\sigma$ in the patches' space in order not to reconstruct the noise. Then, the SVD selects an "average" new direction for each atom, which rejects noise from the dictionary.

Finally, the last formula comes directly from the minimization of (1) when the dictionary \mathbf{D} and the coefficients α_{ij} are fixed and consists of averaging the sparse representation of all overlapping patches. More details are given in [3]. Note also

that finding some other way of performing this average can further improve the results and it is topic of current research. Some ideas along this direction can be found in [43] and [45].

Here, the choice of the parameter C is very important and depends on the dimension of the patches: If \mathbf{w}' is a n -dimensional Gaussian vector, $\|\mathbf{w}'\|_2$ is distributed by the generalized Rayleigh law [46] which leads to the following result:

$$\mathbb{P}(\|\mathbf{w}'\|_2 \leq \sqrt{n}C\sigma) = \frac{1}{\Gamma(\frac{n}{2})} \int_{z=0}^{\frac{nC^2}{2}} z^{\frac{n}{2}-1} e^{-z} dz.$$

In [2], C was tuned empirically to $C = 1.15$ for $n = 8 \times 8 = 64$. Here, we choose the rule

$$\mathbb{P}(\|\mathbf{w}'\|_2 \leq \sqrt{n}C\sigma) = 0.93 \quad (5)$$

which provides a good parameter C for any dimension n .

This approach leads to state-of-the-art grayscale image denoising performance as shown in [2] and [3]. The main challenge to extend this work to color images is to make it able to

capture some correlation between the channels and to reconstruct the image without adding artifacts. Both simple methods, such as denoising each channel separately and denoising directly each patch as a long concatenated RGB vector, fail, respectively, in one of these two challenges. Moreover, for classical color image processing applications such as demosaicing, denoising, and image inpainting [47], spatial and/or spectral nonuniform noise has to be handled. These challenges are addressed next.

IV. SPARSE COLOR IMAGE REPRESENTATION

We now turn to detail the proposed fundamental extensions to the above grayscale framework, and put forward algorithms addressing several different tasks, such as denoising of color images, denoising with nonuniform noise, inpainting small holes of color images, and demosaicing.

A. Denoising of Color Images

The simplest way to extend the K-SVD algorithm to the denoising of color images is to denoise each single channel using a separate algorithm with possibly different dictionaries. However, our goal is to take advantage of the learning capabilities of the K-SVD algorithm to capture the correlation between the different color channels. We will show in our experimental results that a joint method outperforms this trivial plane-by-plane denoising. Recall that although, in this paper, we concentrate on color images, the key extension components here introduced are valid for other modalities of vector-valued images, where the correlation between the planes might be even stronger.

The problem we address is the denoising of RGB color images, represented by a column vector \mathbf{y} , contaminated by some white Gaussian noise \mathbf{w} with a known deviation σ , which has been added to each channel. (As we show in the following, the noise does not have to be uniform across the channels or across the image). Color spaces such as YCbCr, Lab, and other luminance/chrominance separations are often used in image denoising because it is natural to handle the chroma and luma layers differently, and also because the \mathbb{L}_2 -norm in these spaces is more reliable and better reflects the human visual system's perception. However, in this work, we choose to stay with the original RGB space, as any color conversion changes the structure of the noise. Since the OMP step in the algorithm uses an intrinsic hypothesis of a noise with a sphere structure in the patches' space, such an assumption remains valid only in the RGB domain. Other noise geometries do not necessarily guarantee the performance of the OMP in its current form. Another motivation for working in the RGB space is that digital cameras capture images and add noise within this space. One should note also that another extension, which is presented later in this paper, allows to handle nonhomogeneous noise and permits us to, therefore, process images when the noise characteristics are estimated in another color space.

In order to keep a reasonable computational complexity for the color extensions presented in this work, we use dictionaries that are not particularly larger than those practiced in the grayscale version of the algorithm. More specifically, in [2], the authors use dictionaries with 256 atoms and patches of size 8×8 . Applying directly the K-SVD algorithm on (3-D) patches

of size $8 \times 8 \times 3$ (containing the RGB layers) with 256 atoms leads already to substantially better results than denoising each channel separately. However, this direct approach produces artifacts—especially a tendency to reduce the color saturation in the reconstruction. We observe that during the algorithm, the OMP is likely to follow the “gray” axis, which is the axis defined by $r = g = b$ in the RGB color space.

Before proceeding to explain the proposed solution to this color bias and washing effect, let us explain why it happens. As mentioned before, this effect can be seen in the results in [28], although it has not been explicitly addressed there.

First, at the intuitive level, relatively small dictionaries within the order of 256 atoms, for example, are not rich enough to represent the diversity of colors in natural images. Therefore, training a dictionary on a generic database leads to many gray or low chrominance atoms which represent the basic spatial structures of the images. This behavior can be observed in Fig. 2. This result is not unexpected since this global dictionary is aiming at being generic. This predominance of gray atoms in the dictionary encourages the image patches approximation to “follow” the gray axis by picking some gray atoms (via the OMP step, see the following), and this introduces a bias and color washing in the reconstruction. Examples of such color artifacts resulting from global dictionaries are presented in Fig. 3. Using an adaptive dictionary tends to reduce but not eliminate these artifacts. A look at the dictionary in Fig. 4, learned on a specific image instead of a database (global dictionary), shows that the atoms are usually more colored. Our experiments showed that these color artifacts are still present on some image details even with adaptive dictionaries. One might be tempted to solve the above problem by increasing k and, thus, adding redundancy to the dictionary. This, however, is counterproductive in two important ways—the obtained algorithm becomes computationally more demanding and, as the images we handle are getting close in size to the dictionary, over-fitting in the learning process is unavoidable.

We address this color problem by changing the metric of the OMP as will be explained shortly. The OMP is a greedy algorithm that aims to approximate a solution of (2). It consists of selecting at each iteration the best atom from the dictionary, which is the one that maximizes its inner product with the residual (minimizing the error metric), and then updating the residual by performing an orthogonal projection of the signal one wants to approximate onto the vectorial space generated by the previously selected atoms. This orthogonalization is important since it gives more stability and a faster convergence for this greedy algorithm. For details, the reader should refer to [21]–[23] and [48].

An additional, more formal way to explain the lack of colors and the color bias in the reconstruction is to note that the OMP does not guarantee that the reconstructed patch will maintain the average color of the original one. Therefore, the following relationship for the patch ij , $\mathbb{E}(\mathbf{R}_{ij}\mathbf{y}) = \mathbb{E}(\mathbf{R}_{ij}\mathbf{x}_0 + \mathbf{R}_{ij}\mathbf{w}) = \mathbb{E}(\mathbf{R}_{ij}\hat{\mathbf{x}})$, does not necessarily hold. If the diversity of colors is not important enough in the dictionary, the pursuit is likely to follow some other direction in the patches' space. However, with color images and the corresponding increase in the dimensionality, our experiments show that this is clearly the case. To

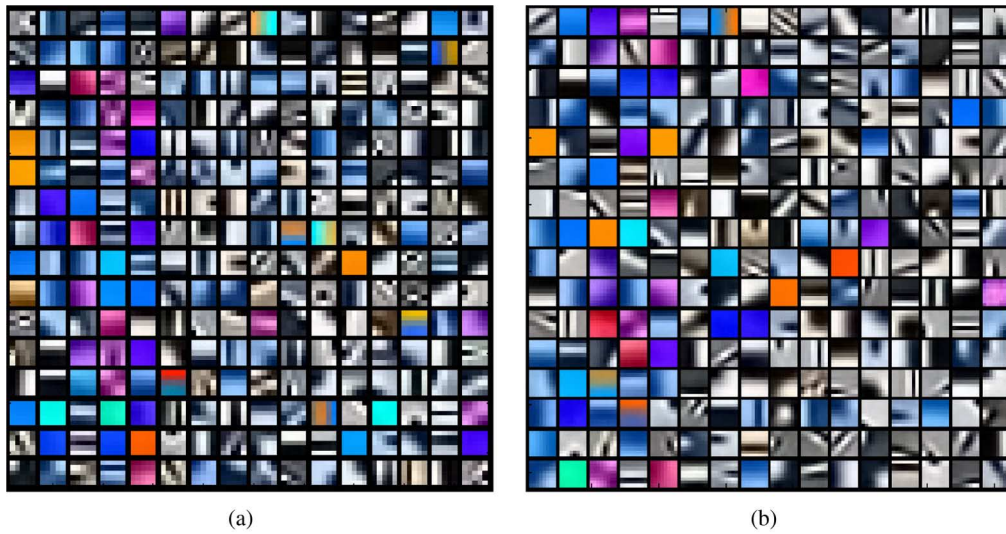


Fig. 2. Dictionaries with 256 atoms learned on a generic database of natural images, with two different sizes of patches. Note the large number of color-less atoms. Since the atoms can have negative values, the vectors are presented scaled and shifted to the $[0,255]$ range per channel: (a) $5 \times 5 \times 3$ patches; (b) $8 \times 8 \times 3$ patches.

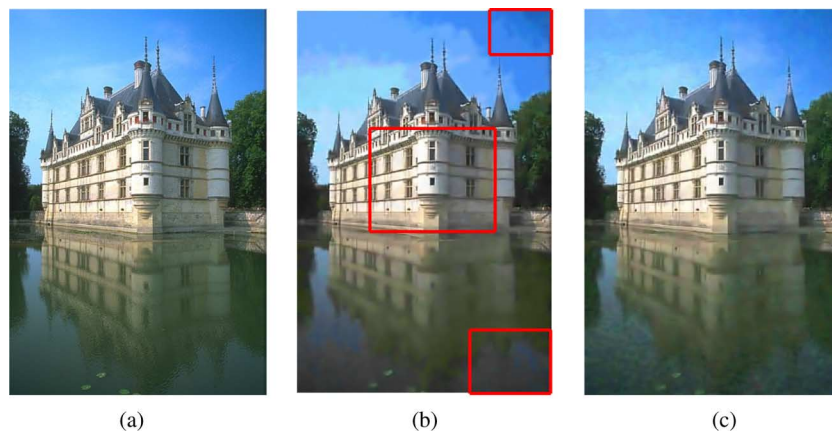


Fig. 3. Examples of color artifacts while reconstructing a damaged version of the image (a) without the improvement here proposed ($\gamma = 0$ in the new metric). Color artifacts are reduced with our proposed technique ($\gamma = 5.25$ in our proposed new metric). Both images have been denoised with the same global dictionary. In (b), one observes a bias effect in the color from the castle and in some part of the water. What is more, the color of the sky is piecewise constant when $\gamma = 0$ (false contours), which is another artifact our approach corrected. (a) Original. (b) Original algorithm, $\gamma = 0$, PSNR = 28.78 dB. (c) Proposed algorithm, $\gamma = 5.25$, PSNR = 30.04 dB.

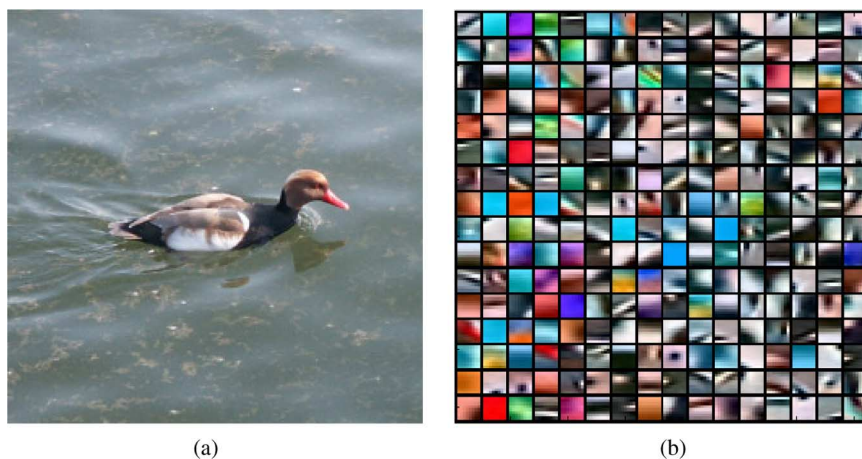


Fig. 4. (a) Training Image; (b) resulting dictionary; (b) is the dictionary learned in the image in (a). The dictionary is more colored than the global one.

address this, we add a force during the OMP which tends to minimize also the bias between the input image and the reconstruction on each channel. Considering that \mathbf{y} and \mathbf{x} are two patches written as column vectors $(R, G, B)^T$, we define a new inner product to be used in the OMP step

$$\langle \mathbf{y}, \mathbf{x} \rangle_\gamma = \mathbf{y}^T \mathbf{x} + \frac{\gamma}{n^2} \mathbf{y}^T \mathbf{K}^T \mathbf{K} \mathbf{x} = \mathbf{y}^T \left(\mathbf{I} + \frac{\gamma}{n} \mathbf{K} \right) \mathbf{x} \quad (6)$$

where

$$\mathbf{K} = \begin{pmatrix} \mathbf{J}_n & 0 & 0 \\ 0 & \mathbf{J}_n & 0 \\ 0 & 0 & \mathbf{J}_n \end{pmatrix}.$$

Here, \mathbf{J}_n is a $n \times n$ matrix filled with ones, and γ is a new parameter which can be tuned to increase or discard this correction. We empirically fixed this parameter to $\gamma = 5.25$. The first term in this equation is the ordinary Euclidean inner product. The second term with the matrix \mathbf{K} computes an estimator of $\mathbb{E}(\mathbf{y})$ and $\mathbb{E}(\mathbf{x})$ on each channel and multiplies them, thereby forcing the selected atoms to take into account the average colors. Examples of results from our algorithm are presented in Fig. 3, with different values for this parameter γ , illustrating the efficiency of our approach in reducing the color artifacts. This correction proved to be crucial in our process, especially for global dictionaries which have a lot of gray atoms as mentioned above. For the tasks we are addressing, the choice of an Euclidean metric is driven in part by the necessity to respect the noise geometry, and it is found to be sufficient to obtain state-of-the-art results. For other purposes, using some metric that is better adapted to the human visual system could be an interesting direction to study. Note that, conceptually, changing the metric in the OMP is straightforward (it can though have consequences in the computational time and convergence theory). Such changes in the dictionary update stage are more difficult and not all metrics are (computationally) easy to handle.

A simple implementation of the above ideas follows from the simple fact that $\mathbf{I} + \gamma/n\mathbf{K} = (\mathbf{I} + a/n\mathbf{K})^T (\mathbf{I} + a/n\mathbf{K})$, with $\gamma = 2a + a^2$. Thereby, scaling each patch and each atom of the dictionary by multiplying them by $\mathbf{I} + a/n\mathbf{K}$, and taking into account a normalization factor, leads to a straightforward implementation of the OMP with the new inner product. This approach permits to easily incorporate the new inner product during the whole OMP process without any further modification. This will effectively change the metric in (3), which becomes

$$\begin{aligned} \forall i,j \hat{\alpha}_{ij} &= \arg \min_{\alpha_{ij}} \|\alpha_{ij}\|_0 \text{ subject to} \\ \left\| \left(\mathbf{I} + \frac{a}{n} \mathbf{K} \right) (\mathbf{R}_{ij} \hat{\mathbf{x}} - \hat{\mathbf{D}} \alpha_{ij}) \right\|_2^2 &\leq n(C\sigma)^2. \end{aligned}$$

We chose not to change the stopping criterion in the OMP to prevent any blurring effect due to the increase in low frequency caused by this scaling. One could wonder why we chose not to change the metric, as well as in (3) and (4), and thereby in (1) (this could effectively be achieved by keeping the scaled image

and dictionaries all across the algorithm). In fact, this metric (inner product) modification is here simply to correct a defect of the OMP when a dictionary can not provide enough diversity in the choice of colors and does not aim at changing the global formulation. Indeed, when computing an adaptive dictionary on an image, when the colors' variability is high, the dictionary suffers from a lack of redundancy and fails at representing this variability. Then, the new metric introduced becomes crucial during the OMP.

To conclude, the basic color denoising algorithm follows the original K-SVD, applied to $\sqrt{n} \times \sqrt{n} \times 3$ patches, with a new metric in the OMP step that explicitly addresses critical color artifacts.

B. Extension to Nonhomogeneous Noise

We now extend the K-SVD algorithm to nonuniform noise. This problem is very important as nonuniform noise across color channels is very common in digital cameras. Spatially nonuniform noise also becomes very important for color demosaicing and inpainting. Here, we address the case of removing white Gaussian noise, but with a different standard deviation for each pixel/color channel, which makes it nonhomogeneous. In this section, we will explain how to use a different metric to overcome this difficulty.

To simplify the presentation, we first consider the case of grayscale images. We denote by $\sigma_p > 0$ the deviation of the noise at the pixel p . We assume that this vector σ is known (this assumption is natural for demosaicing, inpainting, and color dependent noise). In order to be able to use a consistent OMP, we need to have a sphere structure for the noise in the patches' space. This can be explained by the fact that this greedy algorithm aims at finding the sparsest path from the null vector to the vector to approximate in the space generated by the dictionary, by selecting iteratively atoms and reducing the norm of the residual. To prevent the algorithm to retrieve the noise, one has to stop the pursuit when it reaches a high probability of finding the value of the non-noisy patch. As the only aim of the algorithm is to reduce the norm of the residual, the algorithm will provide a maximum efficiency if the stopping criterion is based on a threshold for this norm, thereby it imposes a sphere structure for the noise in the norm's metric.

The first natural idea would be to scale the data so that the deviation of the noise becomes uniform. Scaling the data and then approximating the scaled models leads to loose the image natural structure, which is exactly what we are trying to learn and exploit. Therefore, we need to approximate the nonscaled data using a different metric for each patch where the noise would have a sphere structure. Introducing a vector β composed of weights for each pixel

$$\beta_p = \frac{\min_{p' \in \text{Image}} \sigma_{p'}}{\sigma_p}.$$

It leads us to define a weighted K-SVD algorithm based on a different metric for each patch. Denoting by \otimes the element-wise

Input: \mathbf{E} ($n \times m$ matrix), β ($n \times m$ matrix of weights), iter (number of iterations).

Output: \mathbf{d}_{new} ($n \times 1$ vector), α_{new} ($m \times 1$ vector)

Problem to solve:

$$\arg \min_{\alpha, \|\mathbf{d}\|_2=1} \|\beta \otimes (\mathbf{E} - \mathbf{d}\alpha^T)\|_F^2.$$

Initialization: $\mathbf{d}_{\text{new}} = 0$, $\alpha_{\text{new}} = 0$.

Loop: For i from 1 to iter , use a truncated SVD to solve:

$$(\mathbf{d}_{\text{old}}, \alpha_{\text{old}}) = (\mathbf{d}_{\text{new}}, \alpha_{\text{new}}),$$

$$(\mathbf{d}_{\text{new}}, \alpha_{\text{new}}) = \arg \min_{\alpha, \|\mathbf{d}\|_2=1} \|\beta \otimes \mathbf{E} + (\mathbf{1}_{n \times m} - \beta) \otimes \mathbf{d}_{\text{old}} \alpha_{\text{old}}^T - \mathbf{d} \alpha^T\|_F^2,$$

where $\mathbf{1}_{n \times m}$ is an $n \times m$ matrix filled with ones.

Fig. 5. Weighted rank-one approximation algorithm used for nonuniform noise.

multiplication between two vectors, we aim at solving the following problem, which replaces (2):

$$\begin{aligned} \{\hat{\alpha}_{ij}, \hat{\mathbf{D}}, \hat{\mathbf{x}}\} = & \arg \min_{\mathbf{D}, \alpha_{ij}, x} \lambda \|\beta \otimes (x - y)\|_2^2 \\ & + \sum_{ij} \mu_{ij} \|\alpha_{ij}\|_0 \\ & + \sum_{ij} \|(\mathbf{R}_{ij}\beta) \otimes (\mathbf{D}\alpha_{ij} - \mathbf{R}_{ij}x)\|_2^2. \end{aligned} \quad (7)$$

The OMP step then aims at solving the following equation for the patch ij , instead of (2)

$$\begin{aligned} \forall ij, \hat{\alpha}_{ij} = & \arg \min_{\alpha_{ij}} \|\alpha_{ij}\|_0 \text{ subject to} \\ & \|(\mathbf{R}_{ij}\beta) \otimes (\mathbf{R}_{ij}\hat{\mathbf{x}} - \hat{\mathbf{D}}\alpha_{ij})\|_2^2 \\ & \leq \|\mathbf{R}_{ij}\beta\|_0 (\mathcal{C} \min_p \sigma_p)^2. \end{aligned} \quad (8)$$

The term $\|\mathbf{R}_{ij}\beta\|_0$ here counts the number of pixels in the patch ij without a coefficient β equal to zero which should, therefore, be taken into account. The new inner product (metric) associated with our problem for any vector \mathbf{x} and \mathbf{y} becomes

$$\langle \mathbf{x}, \mathbf{y} \rangle_{ij} = ((\mathbf{R}_{ij}\beta) \otimes \mathbf{x})^T ((\mathbf{R}_{ij}\beta) \otimes \mathbf{y}).$$

Concerning the learning step, the natural approach consists of minimizing for each atom l the following energy, instead of (3)

$$(\hat{\mathbf{d}}_l, \hat{\alpha}^l) = \arg \min_{\alpha^l, \|\mathbf{d}_l\|_2=1} \|\beta^l \otimes (\mathbf{E}_l - \mathbf{d}_l \alpha^l)\|_F^2 \quad (9)$$

where β^l is the matrix whose size is the same as \mathbf{E}_l , and where each column corresponding to an index $[i, j]$ is $\mathbf{R}_{ij}\beta$. This problem is known as a weighted rank-one approximation matrix, is not simple and does not have an unique solution. In [49], Srebro and Jaakkola put forward a simple iterative algorithm which gives an approximated solution of a local minima. We present in Fig. 5 how we applied this algorithm to solve (4), with typically just ten iterations.

Finally, note that the averaging expression, (4), remains the same (modulo the \mathbf{I} and \mathbf{y} in the average, which are weighted by β), since all has been taken into consideration in the previous stages of the algorithm.

Let us now present the model that combines this nonuniform noise handling with the one developed to deal with color artifacts introduced in the previous section. Mixing these two new metrics for each patch ij , we use then the following new inner product (metric) during the OMP step only:

$$\langle \mathbf{x}, \mathbf{y} \rangle_{ij} = ((\mathbf{R}_{ij}\beta) \otimes \mathbf{x})^T \left(\mathbf{I} + \frac{\gamma}{n} \mathbf{K} \right) ((\mathbf{R}_{ij}\beta) \otimes \mathbf{y}).$$

Concerning the learning step, it remains identical to (9), since it does not need any color/bias correction.

To conclude, we have now introduced a new metric that addresses possible color artifacts as well as nonuniform noise. This paves the way for applications beyond color image denoising, and these are described next.

C. Color Image Inpainting

Image inpainting is the art of modifying an image in an undetectable form, and it often refers to the filling-in of holes of missing information in the image [47]. Although our model, with localized patches, is not necessarily an efficient model for filling large holes, since it would require too much computational time and memory to use the K-SVD with a highly redundant dictionary, one can still use this proposed framework for filling holes which are smaller than the size of the atoms. For larger holes, iterative and/or multiscale or texture synthesis methods like in [50] are needed. From a practical point of view, holes smaller than 10×10 pixels were addressed successfully with the algorithm, which is about to be described. A multiscale extension of the K-SVD, which is currently being developed, is able to handle larger holes (results to be reported elsewhere).

The idea for extending the previously described work for inpainting is quite simple. If one considers holes as areas with infinite power noise, this leads to some β_{ij} coefficients equal to 0. To make the model consistent, we also fixed $\sigma = \epsilon > 0$ in

Input: y_m (mosaiced image); D_g (one pre-learned global dictionary).

Demosaic with a global dictionary: y_m using D_g . This gives \hat{x} .

Demosaicing improvement: Apply $Iter$ times:

- *Dictionary Learning:* Learn the dictionary on \hat{x} using a low L patch-sparsity factor starting from D_g . This gives an adaptive dictionary D_a .
- *Demosaicing using joint dictionary:* Demosaic y_m (with the new inner product, derived from non-uniform noise considerations, as defined in the previous section) using the joint dictionary $D_a \cup D_g$. This gives an update of \hat{x} .

Return \hat{x}

Fig. 6. Modified K-SVD algorithm for color image demosaicing. This algorithm combine global dictionaries with data dependent ones learned with a low patch-sparsity factor.

the known areas to prevent infinite β -coefficients. Finally, we consider two possibilities.

- If we have both noise and missing information (holes), we use the model exactly as described above for color image restoration with nonuniform noise.
- If we only have missing information, we change the OMP so that it runs for not more than a fixed number of iterations (this replaces the error-based stopping criteria). Then we use the information of the reconstructed image to fill-in the holes. This approach is faster because it does not force the OMP to fit exactly the known areas and gives similar visual results.

Note also that within the limit of our model, some β -coefficients equal to zero can mask parts of some unnatural atoms, which could end up being used. Therefore, we initialize the algorithm with a global dictionary learned on a large dataset of clean and hole-free images, preventing the use of non-natural patterns like the atoms in the 3-D DCT dictionary, which proved to be inefficient in our experiments for inpainting.

Another possible problem can occur when the matrix of the coefficients β follows a regular pattern. This can lead our algorithm to learn this pattern, absorb it into the dictionary and, thus, overfit. A successful strategy for preventing this is presented in the demosaicing section next, where the holes do form a repetitive pattern (this is much more unusual in ordinary inpainting applications). Note also that with inpainting, we do not have the problem of color artifacts anymore because the constraints of reconstruction are hard (meaning they tend to have a perfect reconstruction on some parts of the image), thereby preventing any bias problem. That is why we chose $\gamma = 0$ in this case.

To conclude, the model for nonuniform noise can be readily exploited for color image inpainting, and examples on this are presented in the experimental section.

D. Color Image Demosaicing

The problem of color demosaicing consists of reconstructing a full resolution image from the raw data produced by a common colored-filtered sensor. Most digital cameras use CCD or CMOS sensors, which are composed of a grid of sensors. One sensor is associated to one pixel and is able to measure

the light energy it receives during a short time. Combined with a color filter [R (red), G (green), or B (blue)], it retrieves the color information of one specific channel. Therefore, often only one color for each pixel is obtained and interpolation of the missing values is necessary. The most used pattern for this is the Bayer pattern, $GRGRGR \dots$ on odd lines and $BGBGBG \dots$ on even ones.

Several algorithms have been developed in recent years to produce high-quality full color images from the mosaic sensor, e.g., [51]–[54]. Although color demosaicing is becoming less relevant with the on-going development of sensor and camera technology, addressing it remains a challenging task that helps to test the effectiveness of different image models and image processing algorithms. We thereby chose to address this problem as a proof of the relevance of our model, helping to show the generality of our framework. The fact that our general model performs as well or even better than state-of-the-art algorithms exclusively developed for image demosaicing, shows the correctness of our approach, and the generality of the sparsity and redundancy concepts, along with the appropriateness of the K-SVD algorithm for learning the dictionary.

We opt to define the problem of demosaicing as an inpainting problem with very small holes which consist of two missing channels per pixel. Considering that we do not need any smoothing effects to get rid of some noise (assuming for the sake of simplicity that the available colors are noise free), nor do we need to inpaint large holes, we chose to restrain the algorithm to use small patches well adapted for retrieving details. We chose, thus, $6 \times 6 \times 3$ patches.

One drawback of our modified K-SVD algorithm with β coefficients is that the presence of a (hole) pattern in the matrix β can lead the algorithm to learn it (the pattern would appear in the dictionary atoms). A simple way of addressing this problem is to use a globally learned dictionary with no-mosaic images. This gives already quite good results that, and as we shall show next, can be further improved.

We introduce the idea of learning the dictionary on a likely image with some artifacts but with a low number of steps during the OMP (low number of atoms will be used to represent the



Fig. 7. Data set used for evaluating denoising experiments.

TABLE I

PSNR RESULTS OF OUR DENOISING ALGORITHM WITH 256 ATOMS OF SIZE $7 \times 7 \times 3$ FOR $\sigma > 10$ AND $6 \times 6 \times 3$ FOR $\sigma \leq 10$. EACH CASE IS DIVIDED IN FOUR PARTS: THE TOP-LEFT RESULTS ARE THOSE GIVEN BY MCAULEY AND AL [28] WITH THEIR “ 3×3 MODEL.” THE TOP-RIGHT RESULTS ARE THOSE OBTAINED BY APPLYING THE GRAYSCALE K-SVD ALGORITHM [2] ON EACH CHANNEL SEPARATELY WITH 8×8 ATOMS. THE BOTTOM-LEFT ARE OUR RESULTS OBTAINED WITH A GLOBALLY TRAINED DICTIONARY. THE BOTTOM-RIGHT ARE THE IMPROVEMENTS OBTAINED WITH THE ADAPTIVE APPROACH WITH 20 ITERATIONS. BOLD INDICATES THE BEST RESULTS FOR EACH GROUP. AS CAN BE SEEN, OUR PROPOSED TECHNIQUE CONSISTENTLY PRODUCES THE BEST RESULTS

σ /PSNR	castle		mushroom		train		horses		kangaroo		Average	
5/34.15	39.90	37.96	38.81	37.51	38.90	35.92	39.25	36.96	38.30	37.51	39.03	37.17
	39.57	40.37	39.61	39.93	37.43	39.76	39.52	40.09	38.54	39.00	38.93	39.83
10/28.13	35.50	33.93	34.16	33.25	33.63	31.04	34.45	32.50	33.14	30.97	34.18	32.24
	35.77	36.24	35.49	35.60	33.68	34.72	35.14	35.43	33.87	34.06	34.79	35.21
15/24.61	33.13	31.70	31.76	30.97	30.50	28.31	31.90	30.24	30.40	28.78	31.54	30.00
	33.4	33.98	33.10	33.18	30.65	31.70	32.53	32.76	31.15	31.30	32.17	32.58
25/20.17	29.98	28.99	28.73	28.37	26.72	25.32	28.84	27.87	27.48	26.69	28.35	27.45
	30.77	31.19	30.21	30.26	27.69	28.16	29.69	29.81	28.31	28.39	29.33	29.56

TABLE II

COMPARISON OF THE PSNR RESULTS ON THE IMAGE “CASTLE” BETWEEN [28] AND WHAT WE OBTAINED WITH $256 \times 6 \times 6 \times 3$ AND $7 \times 7 \times 3$ PATCHES. FOR THE ADAPTIVE APPROACH, 20 ITERATIONS HAVE BEEN PERFORMED. BOLD INDICATES THE BEST RESULT, INDICATING ONCE AGAIN THE CONSISTENT IMPROVEMENT OBTAINED WITH OUR PROPOSED TECHNIQUE

σ /PSNR	[28] 3×3	[28] 5×5	Global $5 \times 5 \times 3$	Adaptive $5 \times 5 \times 3$	Global $6 \times 6 \times 3$	Adaptive $6 \times 6 \times 3$	Global $7 \times 7 \times 3$	Adaptive $7 \times 7 \times 3$
5/34.15	39.90	40.26	39.94	40.16	39.57	40.37	38.20	40.13
10/28.13	35.50	35.91	35.71	36.05	35.77	36.24	35.33	36.25
15/24.61	33.13	33.49	33.27	33.70	33.53	33.93	33.40	33.98
25/20.17	30.00	30.41	30.22	30.80	30.66	31.05	30.77	31.19

patch), in order to prevent any learning of these artifacts (over-fitting). We define then the *patch sparsity* L of the decomposition as this number of steps. The stopping criteria in (2) becomes the number of atoms used instead of the reconstruction error. Using a small L during the OMP permits to learn a dictionary specialized in providing a coarse approximation. Our assumption is that (pattern) artifacts are less present in coarse approximations, preventing the dictionary from learning them. We propose then the algorithm described in Fig. 6. We typically used $L = 2$ to prevent the learning of artifacts and found out that two outer iterations in the scheme in Fig. 6 are sufficient to give satisfactory results, while within the K-SVD, 10–20 iterations are required.

To conclude, in order to address the demosaicing problem, we use the modified K-SVD algorithm that deals with nonuniform noise, as described in previous section, and add to it an adaptive dictionary that has been learned with low patch sparsity in order to avoid over-fitting the mosaic pattern. The same technique can be applied to generic color inpainting as demonstrated in the next section.

V. EXPERIMENTAL RESULTS

We are now ready to present the color image denoising, inpainting, and demosaicing results that are obtained with the proposed framework.

A. Denoising Color Images

The state-of-the-art performance of the algorithm on grayscale images has already been studied in [2]. We now evaluate our extension for color images. We trained some dictionaries with different sizes of atoms $5 \times 5 \times 3$, $6 \times 6 \times 3$, $7 \times 7 \times 3$ and $8 \times 8 \times 3$, on 200 000 patches taken from a database of 15 000 images with the patch-sparsity parameter $L = 6$ (six atoms in the representations). We used the database LabelMe [55] to build our image database. Then we trained each dictionary with 600 iterations. This provided us a set of generic dictionaries that we used as initial dictionaries in our denoising algorithm. Comparing the results obtained with the global approach and the adaptive one permits us to see the improvements in the learning process. We chose to evaluate

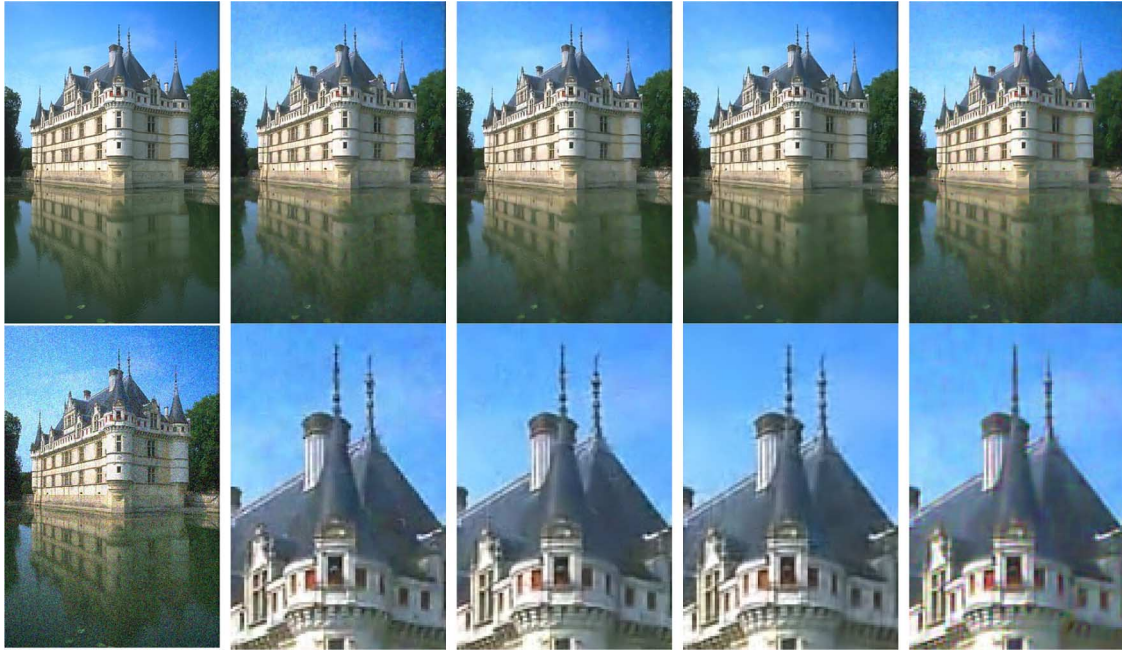


Fig. 8. Results obtained on the castle image. Bottom left image is the noisy image with $\sigma = 25$; top left image is the original one. Then, from left to right, respectively, are presented: the results obtained with patches $5 \times 5 \times 3$; $7 \times 7 \times 3$; $10 \times 10 \times 3$; and the result obtained applying directly the grayscale K-SVD algorithm [2] on each channel separately using patches 8×8 . The bottom row shows a zoomed-in region for each one of the top row images.



Fig. 9. Result obtained by applying our algorithm with $7 \times 7 \times 3$ patches on the mushroom image where a white Gaussian noise of standard deviation $\sigma = 25$ has been added. (a) Original. (b) Noisy. (c) Denoised Image.

our algorithm on some images from the Berkeley segmentation database, [56], presented in Fig. 7. This data selection allows us to compare our results with the relevant work on color image denoising reported in [28], which, as mentioned before, is an extension of [29]. As the raw performance of the algorithm can vary with the different noise realization, the results presented in Tables I and II are averaged over five experiments for each image and each σ . Note that the parameter C has been tuned using (5). For all our experiments, we fixed the number of

iterations J to be 20, which proved to be appropriate for all our results on denoising and inpainting. For detailed experiments about the influence of this parameter, one should refer to [3], where studies for denoising grayscale images are presented. Some visual results are also presented in Fig. 8 in the “castle” image. First of all, one can observe that working on the whole RGB space provides an important improvement when compared to applying grayscale K-SVD, [2], on each channel separately, both in terms of PSNR and visually. Concerning the different

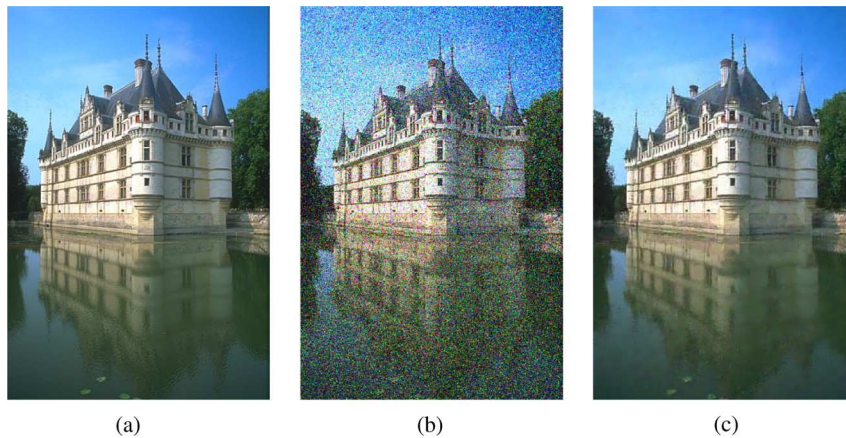


Fig. 10. Example of nonspatially uniform white Gaussian noise. For each image pixel, σ takes a random value from a uniform distribution in $[1; 101]$. The σ values are assumed to be known by the algorithm. Here, the denoising has been performed with $7 \times 7 \times 3$ patches and 20 learning iterations. The initial PSNR was 12.83 dB; the resulting PSNR is 30.18 dB. (a) Original. (b) Noisy. (c) Denoised Image.



Fig. 11. Experiment on a real noisy image. (a) Noisy image. (b) Denoised image.

sizes of patches, small patches are better at retrieving the color of some details whereas large patches are better in flat areas. For example, considering Fig. 8, the sky on the $10 \times 10 \times 3$ image is smoother than on the $5 \times 5 \times 3$, whereas the color of the red curtain behind the windows of the center tower is slightly washed out on the $10 \times 10 \times 3$ image. This motivates in part the learning of multiscale dictionaries as discussed in the conclusion of this paper. Another visual result is presented in Fig. 9 on the “mushroom” image. Besides this, we present an example of denoising a nonuniform noise in Fig. 10, where a white Gaussian noise has been added to each data, but with a different known standard deviation.

The results show that our approach is well adapted to color images. The quality of the results obtained by applying the extended color K-SVD algorithm to the RGB space are significantly better than when denoising each RGB channel separately. Moreover, the algorithm outperforms the most recent work on learning color images reported in [28].

Fig. 11 shows the result of our algorithm on a real noisy image taken from a camera Canon Ixus 900Ti, with a high level ISO setting to obtain a noisy image. Most of the consumer digital cameras today provide pictures that have already been postprocessed, e.g., demosaicing, noise reduction, color correc-

tion and jpeg compression. This proves to be quite destructive in terms of information conservation, especially the inner noise reduction step. This results in a noise that is strongly not white Gaussian, that depends on many factors (intensity of the pixel, color channel, spatial location of the sensor) and is highly spatially correlated. Still, our algorithm, trained on a standard data base and operating with constant variance on each R,G,B channel, produces very good results, pointing to the robustness of the approach. Ideally, of course, our algorithm should operate inside the camera, e.g., as part of the demosaic and attacking the raw data. If to be used as a postprocessing step as here, then to exploit the full power of the framework, a camera-specific dictionary should be learned, and the β matrix needs to be designed to handle this nonuniform noise (both spatially and per channel, since, for example, the blue channel is significantly more noisy). This will lead to even further improvements. The automatic optimization of these algorithm components is subject of future work.

The K-SVD algorithm is relatively fast for denoising. The complexity depends on the number of patches, the sparsity of the decomposition (which depends on σ as well), and the size of the patches. With our experimental software in C++ and a parallel implementation of the OMP, it took approximatively

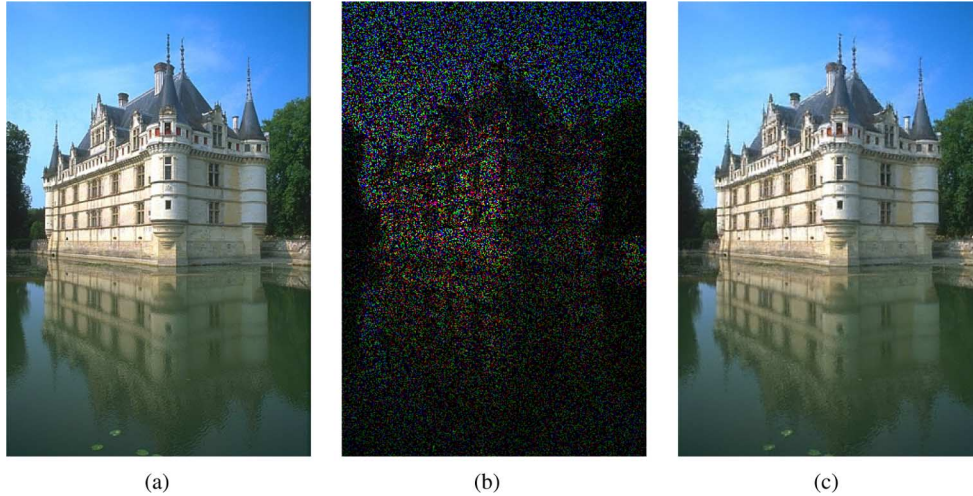


Fig. 12. From left to right: Original image, the image with 80% of data removed, the result of our inpainting using $7 \times 7 \times 3$ patches. The restored image PSNR is 29.65 dB. (a) Original Image. (b) Damaged Image. (c) Restored Image.

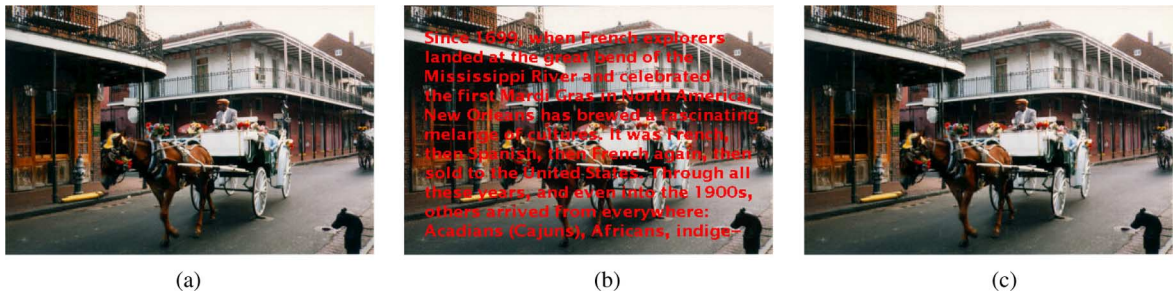


Fig. 13. Inpainting for text removal. (a) Original image. (b) Image with text. (c) Restored image.

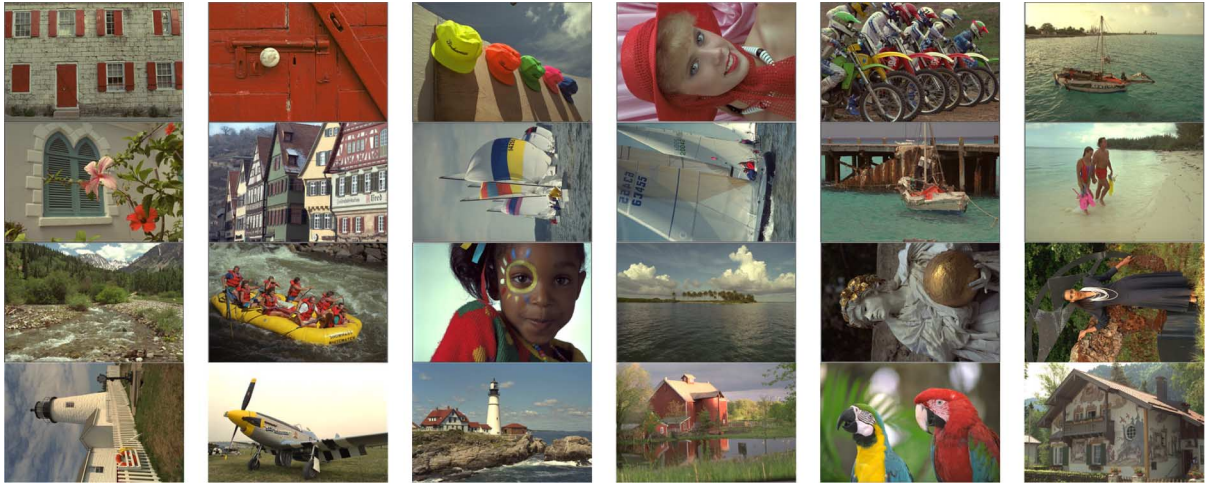


Fig. 14. Kodak image database, images 1 to 24 from left to right and top to bottom, so that the upper left image is n1, and the upper right image is n6.

1.5 s to remove noise with standard deviation $\sigma = 25$ from a $256 \times 256 \times 3$ color image with patches of size $6 \times 6 \times 3$ with a global dictionary (one iteration), and about 32 s. when we performed 20 iterations of the algorithm on a Xeon Quad-core 2.4-GHz processor.

B. Inpainting Color Images

We now present results for inpainting small holes in images. The first example shows the behavior of our algorithm when re-

moving data from the castle image. It is presented in Fig. 12. The second example, Fig. 13, is a classical example of text removal [47] and was used in [29], in order to evaluate their model compared to the pioneer work from [47]. In [29], the Field of Experts model achieves 32.23 dB using their algorithm on the YCbCr space and 32.39 dB on the RGB space. With our model, using $9 \times 9 \times 3$ patches which are large enough to fill in the holes and a dictionary with 512 atoms to ensure the over-completeness, a patch sparsity of 20, and 20 learning iterations, we

TABLE III

COMPARISON OF THE PSNR, IN DECIBELS, FOR DIFFERENT DEMOSAICING ALGORITHMS ON THE KODAK DATA SET. SOME RESULTS ARE TAKEN FROM THE PAPER [51]. BI REFERS TO A SIMPLE BILINEAR INTERPOLATION, THEN, K, AP, OR, SA, CC REFER TO THE ALGORITHMS, RESPECTIVELY, FROM [51], [52], [53], [54], AND [59]. D REFERS TO THE RESULTS OBTAINED WITH A GLOBALLY TRAINED DICTIONARY (600 ITERATIONS WITH $L = 10$ ON 200 000 DIFFERENT $6 \times 6 \times 3$ PATCHES). THEN, DL REFER TO THE RESULT OBTAINED WITH THESE DICTIONARIES AND THE LEARNING PROCESS WE ALREADY DESCRIBED, WITH TWO TIMES 20 LEARNING ITERATIONS WITH A PATCH SPARSITY EQUAL TO 20. BOLD INDICATES THE BEST RESULTS FOR EACH IMAGE, AND THESE ARE MOSTLY ACHIEVED BY OUR ALGORITHM AND SOMETIMES BY THE ONE RECENTLY REPORTED IN [51], WHICH WAS EXPLICITLY DESIGNED FOR HANDLING DEMOSAICING PROBLEMS

Im	BI	K [53]	AP [52]	OR [54]	SA [57]	CC [51]	D	DL
1	26.21	24.64	37.70	34.66	38.32	38.53	36.62	39.37
2	33.08	32.29	39.57	39.22	39.95	40.43	39.34	40.71
3	34.45	33.84	41.45	41.18	41.18	42.54	42.29	43.19
4	33.46	33.45	40.03	38.56	39.55	40.50	40.50	41.29
5	26.51	31.89	37.46	35.25	36.48	37.89	38.29	38.70
6	27.66	35.58	38.50	31.04	39.08	40.03	39.12	40.05
7	33.38	36.36	41.77	41.39	41.50	42.15	42.25	42.83
8	23.39	33.23	35.08	31.04	35.87	36.41	36.01	36.42
9	31.16	39.36	41.72	41.27	41.94	43.04	43.03	43.28
10	32.38	39.22	42.02	40.39	41.80	42.51	42.43	42.70
11	28.96	35.65	39.14	37.42	38.92	39.86	39.64	40.22
12	33.27	39.37	42.51	42.30	42.37	43.45	42.20	43.53
13	23.79	31.07	34.30	31.08	34.91	34.90	35.03	35.29
14	29.05	32.50	35.60	35.58	34.52	36.88	37.27	37.95
15	33.04	34.19	39.53	37.77	38.97	39.78	38.90	40.21
16	31.13	38.12	41.76	41.82	41.60	43.64	42.68	43.62
17	31.80	38.34	41.11	39.06	40.97	41.21	41.74	42.01
18	28.30	32.75	37.45	35.28	37.27	37.49	37.24	37.47
19	27.84	36.50	39.46	38.06	39.96	41.00	40.35	41.27
20	31.51	37.63	40.66	39.05	40.51	41.07	40.49	41.00
21	28.38	36.07	38.66	36.22	38.93	39.12	39.54	39.74
22	30.14	36.00	37.55	36.49	37.67	37.97	38.50	38.87
23	34.83	34.01	41.88	41.34	41.79	42.89	41.00	42.41
24	26.83	32.95	34.78	32.49	34.82	35.04	34.92	35.63
Avg.	30.06	35.21	39.15	37.70	39.12	39.93	39.56	40.32

obtained 32.45 dB. The results from these two models are very similar and both achieve better results than those presented in the original inpainting algorithm [47].

C. Demosaicing

We now present results for demosaicing images. We ran our experiments on the images in Fig. 14, which are taken from the Kodak image database. The size of the images is 512×768 , encoded in RGB with 8 bits per channel. We simulated the mosaic effects using the Bayer pattern, present in Table III the results in terms of PSNR using a globally trained dictionary with atoms of size $6 \times 6 \times 3$, with a patch-sparsity factor $L = 20$, and compare to bilinear interpolation, the results given by Kimmel's algorithm [53], three very recent methods and state-of-the-art results presented in [51].

For some very difficult regions of some images, our generic color image restoration method, when applied to demosaicing, does not give better results than the best interpolation-based methods, which are tuned to track the classical artifacts from

the demosaicing problem. On the other hand, on average, our algorithm performs better than state-of-the-art, improving by a significant 0.39 dB the average result on this standard dataset when compared to the best demosaicing algorithm so far reported. The fact that we did not introduce any special procedure to address the classical demosaicing artifacts and still achieve state-of-the-art results, clearly indicates the power of our framework. Some visual results are presented in Fig. 15.

VI. DISCUSSION AND CONCLUDING REMARKS

In this paper, we introduced a framework for color image restoration, and presented results for color image denoising, inpainting, and demosaicing. The framework is based on learning models for sparse color image representation. The grayscale K-SVD algorithm introduced in [2] and [3] proved to be robust toward the dimensionality increase resulting from the use of color. Following the extensions here introduced, this algorithm learns some correlation between the different R,G,B channels and provides noticeably better results than when modeling each channel separately.

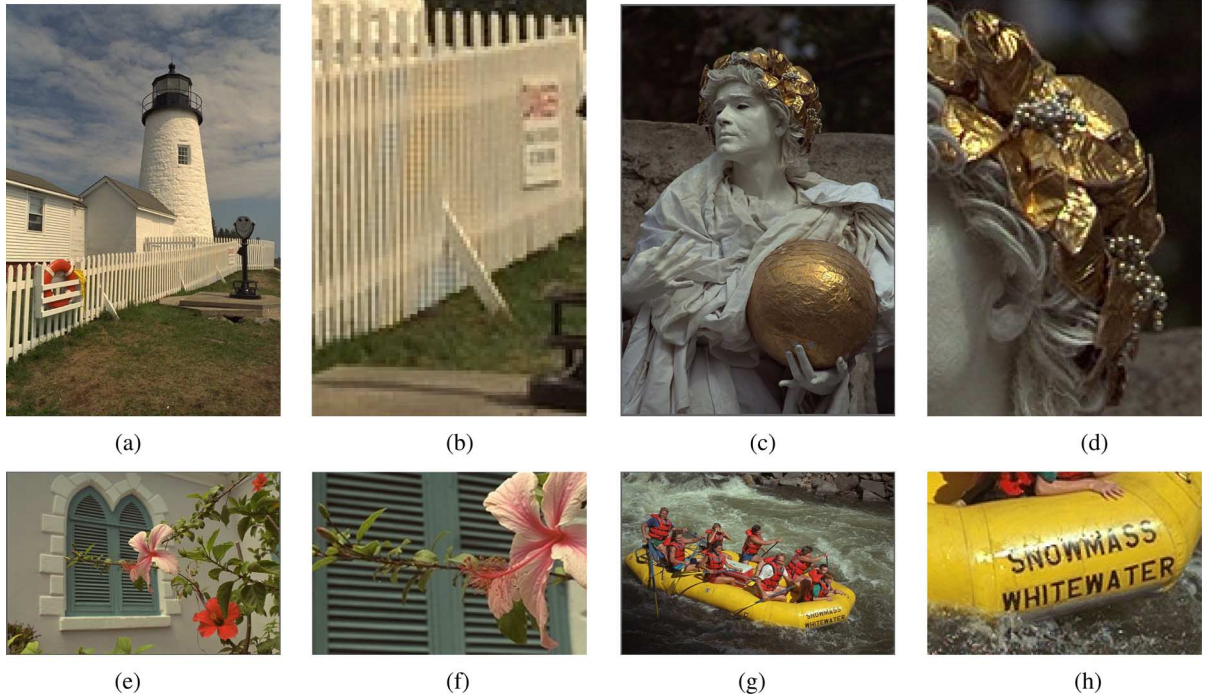


Fig. 15. Examples of demosaicing results.(a) Image 19 restored. (b) Zoomed region. (c) Image 17 restored. (d) Zoomed region. (e) Image 7 restored. (f) Zoomed region. (g) Image 14 restored. (h) Zoomed region.

TABLE IV

PRELIMINARY RESULTS WHEN WORKING WITH TWO SIZES OF PATCHES. THE SVR IS TRAINED ON THE IMAGE OF THE FIRST COLUMN, THEN IT IS USED TO DENOISE THE IMAGE ON THE SECOND COLUMN, WHICH HAS BEEN NOISED WITH A WHITE GAUSSIAN NOISE OF STANDARD DEVIATION $\sigma = 25$. THIRD AND FOURTH COLUMNS INDICATE, RESPECTIVELY, THE PSNR RESULT WHEN DENOISING WITH $5 \times 5 \times 3$ AND $8 \times 8 \times 3$ ATOMS. THE LAST COLUMN PRESENTS THE PSNR RESULT FOR THE JOINT TWO-PATCHES RECONSTRUCTION

Training Image	Noisy image	Small size/PSNR	Big size/PSNR	Result with SVR
Train	Castle	$5 \times 5 \times 3 / 30.74$	$8 \times 8 \times 3 / 31.11$	31.22
Castle	Train	$5 \times 5 \times 3 / 28.04$	$8 \times 8 \times 3 / 28.06$	28.24
Mushroom	Castle	$5 \times 5 \times 3 / 30.65$	$8 \times 8 \times 3 / 31.12$	31.20
Castle	Mushroom	$5 \times 5 \times 3 / 30.10$	$8 \times 8 \times 3 / 30.30$	30.37
Lena	Boat	$5 \times 5 \times 1 / 28.42$	$10 \times 10 \times 1 / 29.37$	29.43
Boat	Lena	$5 \times 5 \times 1 / 29.28$	$10 \times 10 \times 1 / 31.40$	31.41

Although we already obtain state-of-the-art results with the framework presented here, there is still room for future improvement. The artifacts found in images modeled with small patches are often different from those we observed on images modeled with larger patches. Large patches introduce a smoothing effect, giving very good results on flat areas, whereas some details could be blurred. On the other hand, small patches are very good at retrieving details, whereas they might introduce artifacts in flat areas. This motivates us to consider a multiscale version of the K-SVD, and a multiscale learning for image modeling, in general.

We present here a preliminary result in this multiscale direction, working with just two different patch sizes. The idea consists of performing a weighted average between the result obtained with these sizes. The weights are derived from the sparsity of each pixel in the reconstructed images. We define the

pixel-sparsity of one pixel in the reconstructed image as the average sparsity of each patch which contains this pixel.

We consider now grayscale images to simplify the notations. Assume we denoise the image $\mathbf{y} = \mathbf{x} + \mathbf{w}$, following our proposed framework, with one patch size s , obtaining $\hat{\mathbf{x}}^s$, and then with another patch size b , obtaining $\hat{\mathbf{x}}^b$. We want to find the optimal set of weights λ_{ij} for each pixel ij , without knowing the original image \mathbf{x} , such that

$$\lambda_{ij} = \arg \min_{0 \leq \lambda \leq 1} |\mathbf{x}_{ij} - (\lambda \hat{\mathbf{x}}_{ij}^s + (1 - \lambda) \hat{\mathbf{x}}_{ij}^b)|.$$

In order to learn the relationship between the pixel-sparsity and the optimal weights, we use a support vector regression algorithm with a linear kernel [57]. The basic algorithm is presented in Fig. 16. We observed some improvements with this two-scale dictionary, as presented in Table IV, encouraging our

Input: y (noisy image); σ (deviation of the noise); Γ (generic data-set of different images).

Initialization of the database:

- *Creation of the noisy database:* Create $\tilde{\Gamma}$, the noisy version of Γ , where one adds a noise of deviation σ to each image of Γ in order to create the learning database.
- *Denoising the learning database with small patches:* Denoise $\tilde{\Gamma}$ using patches of size $s \times s$. This gives the reconstructed images $\hat{\Gamma}^s$ and the sparsity of each pixel, L_{Γ}^s .
- *Denoising the learning database with large patches:* Denoise $\tilde{\Gamma}$ using patches of size $b \times b$. This gives the reconstructed images $\hat{\Gamma}^b$ and L_{Γ}^b .
- *Computation of the optimal weights:* For each pixel (i, j) of each image m of Γ , denoted by Γ_{ijm} , compute the optimal weight:

$$\lambda_{ijm} = \arg \min_{0 \leq \lambda \leq 1} |\Gamma_{ijm} - (\lambda \hat{\Gamma}_{ijm}^s + (1 - \lambda) \hat{\Gamma}_{ijm}^b)|.$$

Learning the rule between sparsity and optimal weights: Train a Support Vector Regression using L_{Γ}^s and L_{Γ}^b as inputs and the λ_{ijm} as outputs, or eventually one part of this training set.

Denoising process:

- *With small patches:* Denoise y using patches of size $s \times s$. This gives the reconstructed images \hat{x}^s and the sparsity of each pixel, L^s .
- *With large patches:* Denoise y using patches of size $b \times b$. This gives the reconstructed images \hat{x}^b and L^b .
- *Estimation of the optimal weights:* Use the Support Vector Regression to estimate the optimal weights for x using L^s and L^b as inputs. This gives a matrix $\hat{\lambda}$.
- *Weighted averaging:* The output of the algorithm is then the weighted average between \hat{x}^s and \hat{x}^b , with $\hat{\lambda}$ providing the weights.

Fig. 16. K-SVD algorithm for denoising with two sizes of patches.

ongoing research on learning multiscale image models and multiscale sparsity frameworks.

Concerning the computational complexity, the tradeoff between speed and quality can be chosen by the user. Using global dictionaries provides fast results. Inpainting/demosaicing can be performed with a few step in the OMP (parameter L). One great advantage of the K-SVD algorithm which will be more and more important in the future is that it can easily be parallelized. Nowadays, processors are not progressing a lot in the number of sequential operations per second but are multiplying the number of cores inside each chip. Therefore, it has become very im-

portant to design algorithms which can be parallelized. In our case, the OMP can be performed with one patch per processor with maximal efficiency. For the learning step, some libraries like ARPACK provide incomplete singular value decomposition for parallel machines and it is possible to perform the learning of two atoms in parallel if the sets of patches which use them are not overlapping. One strategy to further improve complexity consists of reducing the overlapping of the patches and using only one portion of the patches when working with large images. With patches of size 8×8 and grayscale images 512×512 , a one-fourth reduction proved not to affect the quality of the re-

construction [2]. In our examples, we chose to always use a complete set of overlapping patches, but if one wants to use high definition images with more than one million of pixels, this would be problematic. This is an additional motivation for our current efforts on multiscale frameworks.

We chose to work with natural color images because of their practical relevance and because it is a very convenient data source in order to compare our algorithm with other results available in the literature. One main characteristic we have not discussed in this paper is the capability of this algorithm to be adapted to other types of vectorial data. Future work will consist of testing the algorithm on multichannel images such as LANDSAT. Another interesting future direction consists of learning dictionaries for specific classes of images such as MRI and astronomical data. We strongly believe that this framework could provide cutting edge results in modeling this kind of data, as well.

REFERENCES

- [1] M. Aharon, M. Elad, and A. M. Bruckstein, "The K-SVD: An algorithm for designing of overcomplete dictionaries for sparse representations," *IEEE Trans. Image Process.*, vol. 54, no. 11, pp. 4311–4322, Nov. 2006.
- [2] M. Elad and M. Aharon, "Image denoising via learned dictionaries and sparse representation," presented at the IEEE Computer Vision and Pattern Recognition, New York, Jun. 2006.
- [3] M. Elad and M. Aharon, "Image denoising via sparse and redundant representations over learned dictionaries," *IEEE Trans. Image Process.*, vol. 15, no. 12, pp. 3736–3745, Dec. 2006.
- [4] S. Mallat, *A Wavelet, Tour of Signal Processing, (Wavelet Analysis & Its Applications)*, 2nd ed. New York: Academic, Sep. 1999.
- [5] E. Candes and D. L. Donoho, "Recovering edges in ill-posed inverse problems: Optimality of curvelet frames," *Ann. Statist.*, vol. 30, no. 3, pp. 784–842, Jun. 2002.
- [6] E. Candes and D. L. Donoho, "New tight frames of curvelets and the problem of approximating piecewise C2 images with piecewise C2 edges," *Comm. Pure Appl. Math.*, vol. 57, pp. 219–266, Feb. 2004.
- [7] M. Do and M. Vetterli, "Framing pyramids," *IEEE Trans. Image Process.*, vol. 51, no. 9, pp. 2329–2342, Sep. 2003.
- [8] M. Do and M. Vetterli, G. V. Welland, Ed., *Contourlets, Beyond Wavelets*. New York: Academic, 2003.
- [9] D. Donoho, "Wedgelets: Nearly minimax estimation of edges," *Ann. Statist.*, vol. 27, no. 3, pp. 859–897, Jun. 1998.
- [10] S. Mallat and E. L. Pennec, "Bandelet image approximation and compression," *SIAM J. Multiscale Model. Simul.*, vol. 4, no. 3, pp. 992–1039, Apr. 2005.
- [11] S. Mallat and E. L. Pennec, "Sparse geometric image representation with bandelets," *IEEE Trans. Image Process.*, vol. 14, no. 4, pp. 423–438, Apr. 2005.
- [12] W. T. Freeman and E. H. Adelson, "The design and the use of steerable filters," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 13, no. 9, pp. 891–906, Sep. 1991.
- [13] E. P. Simoncelli, W. T. Freeman, E. H. Adelson, and D. J. Heeger, "Shiftable multi-scale transforms," *IEEE Trans. Inf. Theory*, vol. 38, no. 2, pp. 587–607, Sep. 1992.
- [14] M. W. Marcellin, M. J. Gormish, A. Bilgin, and M. P. Boliek, "An overview of JPEG-2000," in *Proc. Data Compression Conf.*, 2000, pp. 523–544.
- [15] 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, Nov. 2006.
- [16] B. Matalon, M. Elad, and M. Zibulevsky, "Improved denoising of images using modeling of the redundant contourlet transform," in *Proc. SPIE Conf. Wavelets*, Jul. 2005, vol. 5914.
- [17] J. Portilla, V. Strela, M. Wainwright, and E. P. Simoncelli, "Image denoising using scale mixtures of gaussians in the wavelet domain," *IEEE Trans. Image Process.*, vol. 13, no. 4, pp. 496–508, Apr. 2004.
- [18] J.-L. Starck, E. Candes, and D. L. Donoho, "The curvelet transform for image denoising," *IEEE Trans. Image Process.*, vol. 11, no. 6, pp. 670–684, Jun. 2002.
- [19] M. Elad, J.-L. Starck, P. Querre, and D. L. Donoho, "Simultaneous cartoon and texture image inpainting using morphological component analysis (MCA)," *Appl. Comput. Harmon. Anal.*, vol. 19, pp. 340–358, Nov. 2005.
- [20] G. M. Davis, S. Mallat, and Z. Zhang, "Adaptive time-frequency decompositions," *SPIE J. Opt. Eng.*, vol. 33, no. 7, pp. 2183–2191, Jul. 1994.
- [21] S. Mallat and Z. Zhang, "Matching pursuit in a time-frequency dictionary," *IEEE Trans. Signal Process.*, vol. 41, no. 12, pp. 3397–3415, Dec. 1993.
- [22] Y. C. Pati, R. Rezaifar, and P. S. Krishnaprasad, "Orthogonal matching pursuit: Recursive function approximation with applications to wavelet decomposition," presented at the 27th Annu. Asilomar Conf. Signals, Systems, Computers, Nov. 1993.
- [23] S. S. Chen, D. L. Donoho, and M. A. Saunders, "Atomic decomposition by basis pursuit," *SIAM Rev.*, vol. 43, no. 1, pp. 129–59, 2001.
- [24] R. Figueras i Ventura, P. Vanderghynst, P. Frossard, and A. Cavallaro, "Color image scalable coding with matching pursuit," presented at the IEEE Acoust. Speech, Signal Process., May 2004.
- [25] R. Gribonval, C. Fvotte, and E. Vincent, "Performance measurement in blind audio source separation," *IEEE Trans. Speech, Audio, Language Processing*, vol. 14, no. 4, pp. 1462–1469, Jul. 2006.
- [26] R. Gribonval, "Sparse decomposition of stereo signals with matching pursuit and application to blind separation of more than two sources from a stereo mixture," presented at the Int. Conf. Acoust. Speech Signal Processing, Orlando, FL, May 2002.
- [27] S. C. Zhu and D. Mumford, "Prior learning and gibbs reaction-diffusion," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 19, no. 11, pp. 1236–50, Nov. 1997.
- [28] J. McAuley, T. Caetano, A. Smola, and M. O. Franz, "Learning high-order mrf priors of color images," in *Proc. 23rd Int. Conf. Machine Learning*, Jan. 2006, pp. 617–624.
- [29] S. Roth and M. J. Black, "Fields of experts: A framework for learning image priors," presented at the IEEE Computer Vision and Pattern Recognition, San Diego, CA, Jun. 2005.
- [30] R. W. Buccigrossi and E. P. Simoncelli, "Image compression via joint statistical characterization in the wavelet domain," *IEEE Trans. Image Process.*, vol. 8, no. 12, pp. 1688–1701, Dec. 1999.
- [31] E. Haber and L. Tenorio, "Learning regularization functionals," *Inv. Probl.*, vol. 19, pp. 611–626, 2003.
- [32] S. Baker and T. Kanade, "Limits on super-resolution and how to break them," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 9, pp. 1167–1183, Sep. 2002.
- [33] A. A. Efros and T. K. Leung, "Texture synthesis by non-parametric sampling," presented at the IEEE Int. Conf. Computer Vision, Corfu, Greece, Sep. 1999.
- [34] C. Liu, W. Freeman, R. Szeliski, and S. Kang, "Noise estimation from a single image," in *Proc. IEEE Computer Vision and Pattern Recognition*, New York, Jun. 2006, pp. 901–908.
- [35] E. C. P. W. T. Freeman and O. T. Carmichael, "Learning low-level vision," *Int. J. Comput. Vis.*, vol. 40, no. 1, pp. 25–47, 2000.
- [36] A. Criminisi, P. Perez, and K. Toyama, "Region filling and object removal by exemplar-based image inpainting," *IEEE Trans. Image Process.*, vol. 13, no. 9, pp. 1200–1212, Sep. 2004.
- [37] D. Datsenko and M. Elad, "Example-based single document image super-resolution: A global map approach with outlier rejection," *J. Multidimen. Syst. Signal Process.*, no. 2–3, pp. 103–121, Sep. 2007.
- [38] M. Elad and D. Datsenko, "Example-based regularization deployed to super-resolution reconstruction of a single image," *Comput. J.*, vol. 18, no. 2–3, pp. 103–121, Sep. 2007.
- [39] M. F. Tappen, B. C. Russell, and W. T. Freeman, "Exploiting the sparse derivative prior for super-resolution and image demosaicing," presented at the IEEE Workshop Statistical and Computational Theories of Vision, 2003.
- [40] T. Weissman, E. Ordentlich, G. Seroussi, S. Verdu, and M. J. Weinberger, "Universal discrete denoising: Known channel," *IEEE Trans. Inf. Theory*, vol. 15, no. 1, pp. 5–28, Jan. 2005.
- [41] A. Buades, B. Coll, and J.-M. Morel, "A non-local algorithm for image denoising," presented at the IEEE Computer Vision and Pattern Recognition, San Diego, CA, Jun. 2005.
- [42] A. Buades, B. Coll, and J. M. Morel, "A review of image denoising algorithms, with a new one," *SIAM J. Multiscale Model. Simul.*, vol. 4, no. 2, pp. 490–530, 2005.
- [43] C. Kervrann and J. Boulanger, "Optimal spatial adaptation for patch-based image denoising," *IEEE Trans. Image Process.*, vol. 15, no. 10, pp. 2866–2878, Oct. 2006.

- [44] M. Mahmoudi and G. Sapiro, "Fast image and video denoising via non-local means of similar neighborhoods," *IEEE Signal Process. Lett.*, vol. 2, no. 12, pp. 839–842, Dec. 2005.
- [45] O. G. Guleryuz, "Weighted overcomplete denoising," to be published.
- [46] K. S. Miller, *Multidimensional Gaussian Distributions*. New York: Wiley, 1964.
- [47] M. Bertalmio, G. Sapiro, V. Caselles, and C. Ballester, "Image in-painting," in *Proc. 27th Annu. Conf. Computer Graphics and Interactive Techniques*, 2000, pp. 417–424.
- [48] J. A. Tropp, "Greed is good: Algorithmic results for sparse approximation," *IEEE Trans. Inf. Theory*, vol. 50, no. 10, pp. 2231–2242, Oct. 2004.
- [49] N. Srebro and T. Jaakkola, "Weighted low-rank approximations," in *Proc. ICML*, 2003, pp. 720–727.
- [50] A. Criminisi, P. Perez, and K. Toyama, "Object removal by exemplar-based inpainting," presented at the IEEE Computer Vision and Pattern Recognition, Madison, WI, Jun. 2003.
- [51] K.-H. Chung and Y.-H. Chan, "Color demosaicing using variance of color differences," *IEEE Trans. Image Process.*, vol. 15, no. 10, pp. 2944–2955, Oct. 2006.
- [52] B. K. Gunturk, Y. Altunbask, and R. M. Mersereau, "Color plane interpolation using alternating projections," *IEEE Trans. Image Process.*, vol. 11, no. 9, pp. 997–1013, Sep. 2002.
- [53] R. Kimmel, "Demosaicing: Image reconstruction from color ccd samples," *IEEE Trans. Image Process.*, vol. 8, no. 9, pp. 1221–1228, Sep. 1999.
- [54] D. D. Muresan and T. W. Parks, "Demosaicing using optimal recovery," *IEEE Trans. Image Process.*, vol. 14, no. 2, pp. 267–278, Feb. 2005.
- [55] B. C. Russell, A. Torralba, K. P. Murphy, and W. T. Freeman, "Labelme: A database and web-based tool for image annotation," Tech. Rep., Massachusetts Inst. Technol. Lab Memo AIM-2005-025, Sep. 2005.
- [56] 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. IEEE Int. Conf. Computer Vision*, Jul. 2001, vol. 2, pp. 416–423.
- [57] A. J. Smola and B. Schölkopf, "A tutorial on support vector regression," *Statist. Comput.*, vol. 14, no. 3, pp. 199–222, 2004.
- [58] B. A. Olshausen and D. J. Field, "Sparse coding with an overcomplete basis set: A strategy employed by V1?," *Vis. Res.*, vol. 37, no. 23, pp. 3311–3325, Dec. 1997.
- [59] X. Li, "Demosaicing by successive approximations," *IEEE Trans. Image Process.*, vol. 14, no. 2, pp. 267–278, Feb. 2005.



Julien Mairal (M'08) was born in Clermont-Ferrand, France, in 1982. He received the B.Sc. degree from the Ecole Polytechnique, Paris, in 2005, and from the Ecole Nationale Supérieure des Télécommunications, Paris, in 2007, and the M.Sc. degree from the Ecole Polytechnique and the Ecole Normale Supérieure, Cachan, in 2006. He is currently pursuing the Ph.D. degree under the supervision of J. Ponce and A. Zisserman.

In 2007, he became "Ingénieur du Corps des Télécommunications". During his studies, he worked in the INRIA project Odyssey, Paris, under the supervision of R. Keriven in 2005, and in the Department of Electrical and Computer Engineering, in the University of Minnesota, Minneapolis, under the supervision of G. Sapiro in 2006. Recently, he joined the INRIA project Willow, Paris.



Michael Elad (M'05) received the B.Sc., M.Sc. (supervision by Prof. D. Malah), and D.Sc. (supervision by Prof. A. Feuer) from the Department of Electrical Engineering, The Technion—Israel Institute of Technology, Haifa, in 1986, 1988, and 1997, respectively.

From 1988 to 1993, he served in the Israeli Air Force. From 1997 to 2000, he was with Hewlett-Packard Laboratories as an R&D Engineer. From 2000 to 2001, he headed the research division at Jigami Corporation, Israel. During the years 2001 to 2003, he spent a postdoctorate period as a Research Associate with the Computer Science Department, Stanford University (SCCM program), Stanford, CA. In September 2003, he returned to The Technion, assuming a tenure-track assistant professorship position in the Department of Computer Science. In May 2007, Michael was tenured to an Associate Professor. He works in the field of signal and image processing, specializing particularly in inverse problems, sparse representations, and overcomplete transforms.

Dr. Elad received The Technion's Best Lecturer Award five times (1999, 2000, 2004, 2005, and 2006). He received the Solomon Simon Mani award for excellence in teaching in 2007. He is also the recipient of the Guttwirth and the Wolf fellowships. He is currently an Associate Editor for the IEEE TRANSACTIONS ON IMAGE PROCESSING.



Guillermo Sapiro (M'94) was born in Montevideo, Uruguay, on April 3, 1966. He received the B.Sc. (*summa cum laude*), M.Sc., and Ph.D. degrees from the Department of Electrical Engineering, The Technion—Israel Institute of Technology, Haifa, in 1989, 1991, and 1993, respectively.

After postdoctoral research at the Massachusetts Institute of Technology, Cambridge, he became a Member of Technical Staff at the research facilities of HP Labs, Palo Alto, CA. He is currently with the Department of Electrical and Computer Engineering, University of Minnesota, Minneapolis, where he holds the position of Distinguished McKnight University Professor. He works on differential geometry and geometric partial differential equations, both in theory and applications in computer vision, computer graphics, medical imaging, and image analysis. He has authored and coauthored numerous papers in these areas and has written the book *Geometric Partial Differential Equations and Image Processing* (Cambridge University Press, 2001).

Dr. Sapiro is a member of SIAM. He has coedited special issues of the IEEE TRANSACTIONS ON IMAGE PROCESSING and the *Journal of Visual Communication and Image Representation*. He was awarded the Guttwirth Scholarship for Special Excellence in Graduate Studies in 1991, the Ollendorff Fellowship for Excellence in Vision and Image Understanding Work in 1992, the Rothschild Fellowship for postdoctoral studies in 1993, the Office of Naval Research Young Investigator Award in 1998, the Presidential Early Career Awards for Scientist and Engineers (PECASE) in 1998, and the National Science Foundation Career Award in 1999.