SPARSE REPRESENTATION FOR COLOR IMAGE RESTORATION

By

Julien Mairal

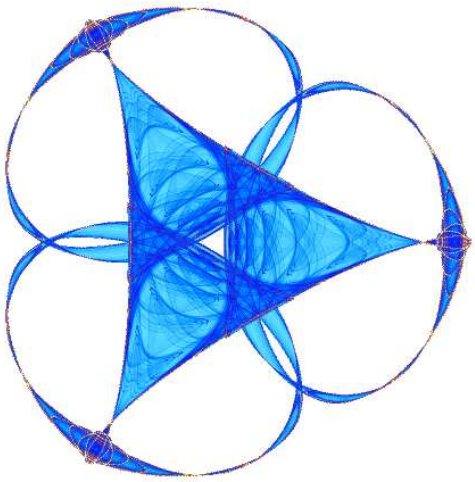
Michael Elad

and

Guillermo Sapiro

IMA Preprint Series # 2139

(October 2006)



INSTITUTE FOR MATHEMATICS AND ITS APPLICATIONS

UNIVERSITY OF MINNESOTA

400 Lind Hall

207 Church Street S.E.

Minneapolis, Minnesota 55455–0436

Phone: 612/624-6066 Fax: 612/626-7370

URL: http://www.ima.umn.edu

Report Documentation Page

Form Approved
OMB No. 0704-0188

1. REPORT DATE	2. REPORT TYPE	3. DATES COVERED
OCT 2006		**00-00-2006 to 00-00-2006**

4. TITLE AND SUBTITLE	5a. CONTRACT NUMBER
Sparse Representation for Color Image Restoration (PREPRINT)	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	5d. PROJECT NUMBER
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **University of Minnesota,Institute for Mathematics and its Applications,207 Church Street SE,Minneapolis,MN,55455-0436**	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR'S ACRONYM(S)
	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES

14. 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 gray-scale 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.

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			
unclassified	**unclassified**	**unclassified**	**Same as Report (SAR)**	**31**	

Sparse Representation for Color Image Restoration

Julien Mairal,[1*] Michael Elad,[2] and Guillermo Sapiro[3]

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 gray-scale image processing tasks. In this paper we address the problem of learning dictionaries for color images and extend the K-SVD-based gray-scale image denoising algorithm that appears in [2]. This work puts forward ways for handling non-homogeneous 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.

EDICS Category: COL-COLR (Color processing)

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.

Work partially supported by NSF, ONR, NGA, DARPA, and the McKnight Foundation.

1. Department of Electrical and Computer Engineering, University of Minnesota, 200 Union Street SE Minneapolis, MN 55455 USA, Email: julien.mairal@m4x.org. Phone: +1-612-6251343. Fax: +1-612-6254583.

2. Department of Computer Science, The Technion – Israel Institute of Technology, Haifa 32000 Israel, Email: elad@cs.technion.ac.il. Phone: +972-4-829-4169. Fax: +972-4-829-4353.

3.Department of Electrical and Computer Engineering, University of Minnesota, 200 Union Street SE Minneapolis, MN 55455 USA, E-mail: guille@ece.umn.edu. Phone: +1-612-6251343. Fax: +1-612-6254583.

The *Sparseland* model is an emerging and powerful method to describe signals based on the sparsity and redundancy of their representations [3], [2]. 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

$$\forall \mathbf{x} \in \Gamma,\ \exists \alpha \in \mathbb{R}^k \quad \text{such that} \quad \mathbf{x} \approx \mathbf{D}\alpha \ \text{ and } \ \|\alpha\|_0 \ll n.$$

The notation $\|\cdot\|_0$ is the ℓ^0-norm which counts the number of non-zero 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], [16], [17], [18], inpainting [19], and more. Common to all these pre-defined 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 pre-defined ones as described above, that leads to sparse representations on training signals drawn from Γ. This algorithm uses either Orthogonal Matching Pursuit (OMP) [20], [21], [22], or Basis Pursuit (BP), [23], as part of its iterative procedure for learning the dictionary. The follow-up work reported in [3], [2] proposes a novel and highly effective image denoising algorithm for the removal of additive white Gaussian noise with gray-scale images. Their proposed method includes the use of the K-SVD for learning the dictionary from the noisy image directly.

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 non-homogeneous 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 2 we describe prior art on example-based image denoising (and general image processing) methods. Section 3 is devoted to a brief description of the K-SVD-based gray-scale image denoising algorithm as proposed in [2]. Section 4 describes the novelties offered in this paper – the extension to color images, the handling of color artifacts, and finally the treatment of non-homogeneous noise, along with its relation to demosaicing and inpainting. In Section 5 we provide various experiments that demonstrate the effectiveness of the proposed algorithms. Section 6 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 non-parametric 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 [24] and later in [25], [26]; the wavelet based image prior as appears in [27]; the Tikhonov regularization proposed by Haber and Tenorio [28]; the super-resolution approach adopted by Baker and Kanade [29]; and the recent K-SVD denoising as described in [3], [2].

The alternative path, a non-parametric learning, use image examples directly within the reconstruction process, as practiced by Efros and Leung for texture synthesis [30]; by Freeman et. al. for super-resolution [31], [32]; and by several follow-up works [33], [34], [35], [36] 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 [37] 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 Non-Local-Means [38], [39] and related works [40], [41]. 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 dictionary 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 non homogeneous 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 [25]. This work builds on the *Field of Experts*, as developed in [26], 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 [26]. Color artifacts are not mentioned, and indeed, the shown results demonstrate a phenomenon we have experienced too, of a tendency to wash-out 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. The gray-scale K-SVD denoising algorithm

In [3], [2], Aharon and Elad present a K-SVD-based algorithm for denoising of gray-scale 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:

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

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 it's 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 Equation (1) introduces the likelihood force that demands a proximity between $\hat{\mathbf{x}}$ 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 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 [3], [2] put forward an iterative method that incorporates the K-SVD algorithm, as presented in [1]. Figure 1 presents this image denoising algorithm in details.

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 $\hat{\mathbf{D}} = (\hat{\mathbf{d}}_l)_{l \in 1 \ldots k}$ 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

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

- *Dictionary Update:* Fix all $\hat{\alpha}_{ij}$, and for each atom $l \in 1, 2, \ldots, k$ 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:

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

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

Averaging: Perform a weighted average:

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

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

One can notice that different steps in this algorithm reject the noise. First, the Matching Pursuit (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 performs an average between the representation of overlapping patches.

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 [42] 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

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

which provides a good parameter C for any dimension n.

This approach leads to state-of-the-art gray-scale image denoising performance as shown in [3], [2]. 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, [43], spatial and/or spectral non-uniform 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 gray-scale framework, and put forward algorithms addressing several different tasks, such as denoising of color images, denoising with non-uniform 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.

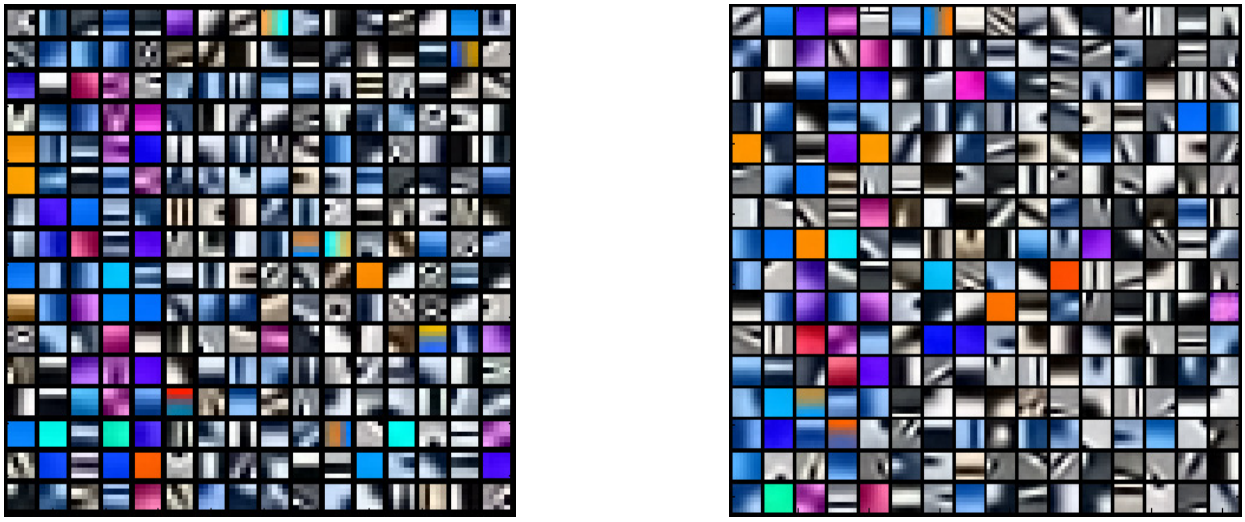
The problem we address is the denoising of RGB color images, represented by a column vector $\mathbf{y}$, contaminated by some white Gaussian noise $\mathbf{w}$ with a known deviation σ, which has been added to each channel. (As we show below, the noise does not have to be uniform across the channels 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 assumption remains valid only in the RGB domain. Other noise geometries do not necessarily guarantee the performance of the OMP.

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 gray-scale 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 (three dimensional) 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 [25], 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 Figure 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

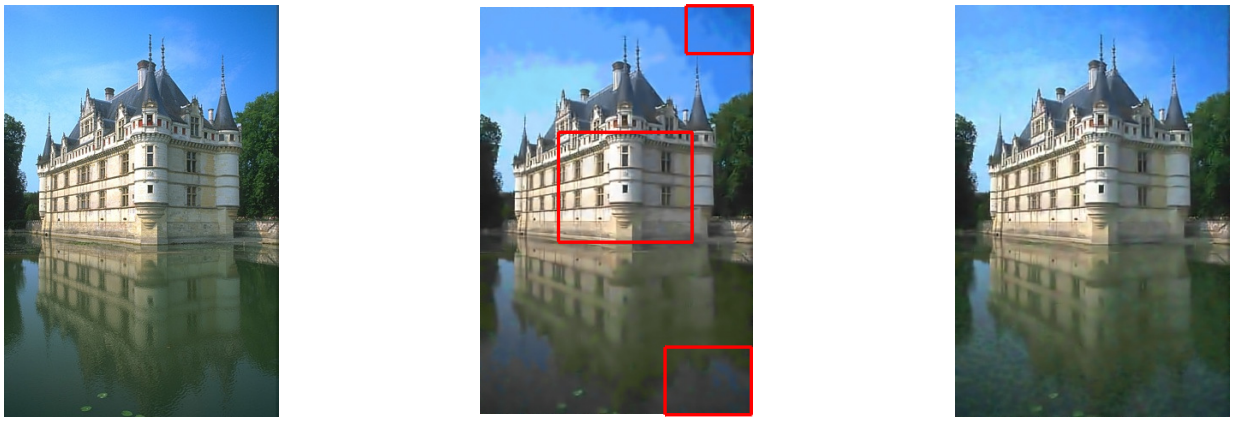
"follow" the gray axis by picking some gray atoms (via the OMP step, see below), and this introduces a bias and color washing in the reconstruction. Examples of such color artifacts resulting from global dictionaries are presented in Figure 3. Using an adaptive dictionary tends to reduce but not eliminate these artifacts. A look at the dictionary in Figure 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 counter productive 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 in unavoidable.



(a) $5 \times 5 \times 3$ patches (b) $8 \times 8 \times 3$ patches

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.*

We address this color problem by changing the metric of the OMP as it will be explained shortly. The OMP is a greedy algorithm that aims to approximate a solution of Equation (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



(a) Original

(b) Original algorithm, $\gamma = 0$, PSNR=28.78 dB

(c) Proposed algorithm, $\gamma = 5.25$, PSNR=30.04 dB

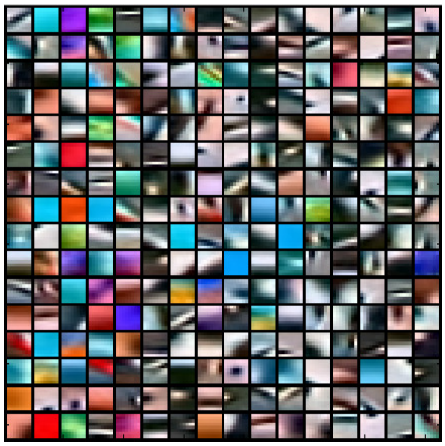
Fig. 3. *Examples of color artifacts while reconstructing a damaged version of the image 3(a) without the improvement here proposed ($\gamma = 0$ in the new metric). Color artifacts are reduced with our proposed technique ($\gamma = 5.25$ in our proposed new metric). Both images have been denoised with the same global dictionary. In 3(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.*

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 [23], [22], [21].

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. But with color images and the corresponding increase in the dimensionality, our experiments show that this is clearly the case. To 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:

$$< \mathbf{y}, \mathbf{x} >_\gamma = \mathbf{y}^T\mathbf{x} + \frac{\gamma}{n^2}\mathbf{y}^T\mathbf{K}^T\mathbf{K}\mathbf{x} = \mathbf{y}^T(\mathbf{I} + \frac{\gamma}{n}\mathbf{K})\mathbf{x}, \tag{6}$$

(a) Training Image



(b) Resulting dictionary

Fig. 4. *Figure 4(b) is the dictionary learned on the image 4(a). The dictionary is more colored than the global one.*

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 on Figure 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.

A simple implementation of the above ideas follows from the simple fact that $\mathbf{I} + \frac{\gamma}{n}\mathbf{K} = (\mathbf{I} + \frac{a}{n}\mathbf{K})^T(\mathbf{I} + \frac{a}{n}\mathbf{K})$, with $\gamma = 2a + a^2$. Thereby, scaling each patch and each atom of the dictionary by multiplying them by $\mathbf{I} + \frac{a}{n}\mathbf{K}$, and taking into account a normalization factor, leads to a straightforward implementation of the OMP with the new inner product. This approach effectively changes the metric in Equation (2). 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 in equations (3) and (4), and thereby in Equation (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.

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 non-homogeneous noise

We now extend the K-SVD algorithm to non-uniform noise. This problem is very important as non-uniform noise across color channels is very common in digital cameras. Spatially non-uniform noise also becomes very important for color demosaicing and inpainting.

To simplify the presentation, we first consider the case of gray-scale 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 non-scaled 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 multiplication between two vectors, we aim at solving the following problem, which replaces Equation (1):

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

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

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

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:

$$< \mathbf{x}, \mathbf{y} >_{ij} \quad = \quad ((\mathbf{R}_{ij}\beta)\otimes\mathbf{x})^T((\mathbf{R}_{ij}\beta)\otimes\mathbf{y}).$$

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

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

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 one-rank approximation matrix, is not simple and has not an unique solution. In [44], Srebro and Jaakkola put forward a simple iterative algorithm which gives an approximated solution of a local minimum. Nevertheless, this algorithm requires a SVD for each of its iterations, being relatively complex. We chose instead for each atom l to perform a one-rank approximation with a single truncated SVD of the most consistent term, which is composed of the contribution of one atom in the reconstruction plus its weighted residual:

$$(\hat{\mathbf{d}}_l, \hat{\alpha}^l) = \arg\min_{\alpha,||\mathbf{d}||_2=1}||\mathbf{E}_l^\beta-\mathbf{d}\alpha||_F^2, \quad (10)$$

where $\mathbf{E}_l^\beta$ is the matrix whose columns are

$$\mathbf{e}_l^\beta(ij) = \underbrace{(\mathbf{R}_{ij}\beta)\otimes(\mathbf{R}_{ij}\hat{\mathbf{x}}-\hat{\mathbf{D}}\hat{\alpha}_{ij})}_{\text{weighted residual}}+\underbrace{\hat{\mathbf{d}}_l\hat{\alpha}_{ij}(l)}_{\text{contribution}}. \quad (11)$$

Note that we can rewrite this expression as

$$\mathbf{e}_l^\beta(ij) = (\mathbf{R}_{ij}\beta)\otimes(\mathbf{R}_{ij}\hat{\mathbf{x}}-\hat{\mathbf{D}}\hat{\alpha}_{ij}+\hat{\mathbf{d}}_l\hat{\alpha}_{ij}(l))+(\mathbf{1}_n-\mathbf{R}_{ij}\beta)\otimes\hat{\mathbf{d}}_l\hat{\alpha}_{ij}(l),$$

where $\mathbf{1}_n$ denotes a column vector filled with ones. This is actually the first iteration of the algorithm put forward by Srebro and Jaakkola in [44]. One can interpret the first term $(\mathbf{R}_{ij}\beta) \otimes (\mathbf{R}_{ij}\hat{\mathbf{x}} - \hat{\mathbf{D}}\hat{\alpha}_{ij} + \hat{\mathbf{d}}_l\hat{\alpha}_{ij}(l))$ as the relevant information we dispose to make the atoms evolve, whereas $(\mathbf{1}_n - \mathbf{R}_{ij}\beta) \otimes \hat{\mathbf{d}}_l\hat{\alpha}_{ij}(l)$ can be viewed as a balancing term, completing the equation and permitting to perform the SVD.

Finally, note that the averaging expression, Equation (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 non-uniform 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:

$$< \mathbf{x}, \mathbf{y} >_{ij} \quad = \quad ((\mathbf{R}_{ij}\beta) \otimes \mathbf{x})^T (I + \frac{\gamma}{n}\mathbf{K})((\mathbf{R}_{ij}\beta) \otimes \mathbf{y})$$

Concerning the learning step, it remains identical to Equation (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 non-uniform 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 [43]. Although sparsity, as practiced here with localized patches, is not necessarily an efficient model for filling large holes, since it would intrinsically lead to a lack of details, one can still use it for filling small holes, as long as their sizes are smaller than the size of the atoms. For larger holes, iterative and/or multiscale or texture synthesis methods like in [45] are needed.

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 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 non-uniform 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 3D-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 non-uniform 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 the missing

values is necessary. The most used pattern for this is the Bayer pattern, *GRGRGR...* on odd lines and *BGBGBG...* on even ones.

Several algorithms have been developed in recent years to produce high quality full color images from the mosaic sensor, e.g., [46], [47], [48], [49]. 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), neither we need to inpaint large holes, we chose to restrain the algorithm to use small patches well adapted for retrieving details. We chose thus $5 \times 5 \times 3$ 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, 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 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 Equation (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 Figure 5. We typically used $L = 2$ to prevent the learning of artifacts and found out that two outer iterations

in the scheme in Figure 5 are sufficient to give satisfactory results, while within the K-SVD, 10-20 iterations are required.

Input: $\mathbf{y}_m$ (mosaiced image); $\mathbf{D}_g$ (one pre-learned global dictionary).

Demosaic with a global dictionary: $\mathbf{y}_m$ using $\mathbf{D}_g$. This gives $\hat{\mathbf{x}}$.

Demosaicing improvement: Apply $Iter$ times:

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

Return $\hat{\mathbf{x}}$

Fig. 5. *Modified K-SVD algorithm for color image demosaicing. This algorithm combine global dictionaries with data dependent ones learned with a low patch-sparsity factor.*

To conclude, in order to address the demosaicing problem, we use the modified K-SVD algorithm that deals with non-uniform 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 gray-scale 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 200000 patches taken from a database of 15000 images with the patch-sparsity parameter $L = 6$ (6 atoms in the representations). We used the database LabelMe, [50], 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 to see the improvements in the learning process. We chose to evaluate our algorithm on some images from the Berkeley Segmentation database, [51], presented in Figure 6. This data selection allows us to compare our results with the relevant work on color image denoising reported in [25], which as mentioned before, is an extension of [26]. As the raw performance of the algorithm can vary with the different noise realization, the results presented in Table I and Table II are averaged over 5 experiments for each image and each σ. Note that the parameter C has been tuned using Equation (5). Some visual results are also presented on Figure 7 on the "castle" image. First of all, one can observe that working on the whole RGB space provides an important improvement when compared to applying gray-scale K-SVD, [2], on each channel separately, both in terms of PSNR and visually. Concerning the different sizes of patches, small patches are better at retrieving the color of some details whereas large patches are better in flat areas. For example, taking the example of Figure 7, 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 Figure 8 on the "mushroom" image. Besides this, we present an example of denoising a non uniform noise on Figure 9, where a white Gaussian noise has been added to each data, but with a different known standard deviation.

Fig. 6. *Data set used for evaluating denoising experiments*

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 [25].

The K-SVD algorithm is relatively fast for denoising. The complexity depends on the number

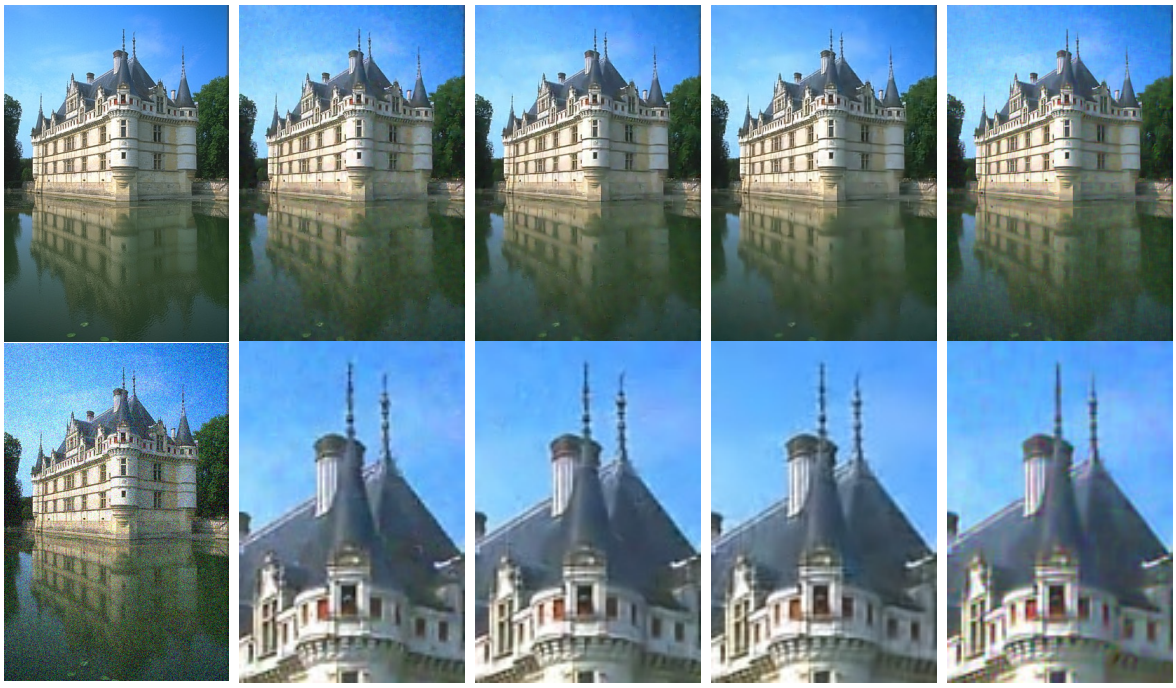


Fig. 7. *Results obtained on the castle image. Bottom left image is the noisy image with* $\sigma = 25$*, top left image is the original one. Then, from left to right are respectively presented: the results obtained with patches* $5 \times 5 \times 3$*,* $7 \times 7 \times 3$*,* $10 \times 10 \times 3$*, and the result obtained applying directly the gray-scale 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.*

(a) Original (b) Noisy (c) Denoised Image

Fig. 8. *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.*

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 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 [25] with their "*3×3 *model." The top-right results are those obtained by applying the gray-scale 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.*

of patches, the sparsity of the decomposition (which depends on σ as well), and the size of the patches. With our experimental implementation in C++, it took approximatively 4.4s to remove noise with standard deviation $\sigma = 25$ from a $256 \times 256 \times 3$ color image with patches of size $5 \times 5 \times 3$ with a global dictionary (one iteration), and about 1 min. 40 sec. when we performed 20 iterations of the algorithm on a Pentium-M 1.73GHz processor.

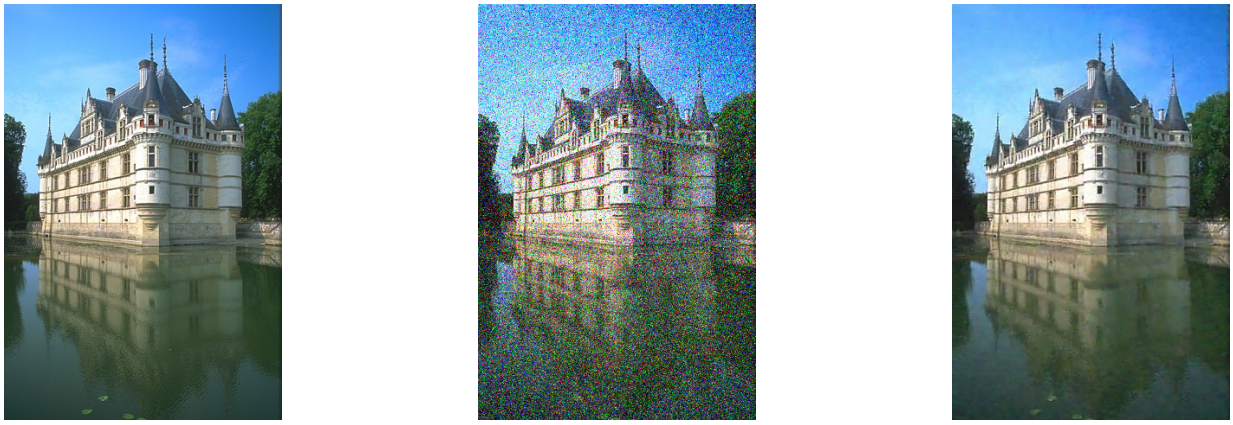
B. Inpainting color images

We now present results for inpainting small holes in images. The first example shows the behavior of our algorithm when removing data from the castle image. It is presented on Figure 10. The second example, Figure 11, is a classical example of text removal, [43], and was used in [26] in order to evaluate their model compared to the pioneer work from [43]. In [26], 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 obtained 32.45 dB. The results from these two models are very

σ/PSNR	[25] 3×3	[25] 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**

TABLE II

Comparison of the PSNR results on the image "castle" between [25] and what we obtained with 256 $6 \times 6 \times 3$ *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.*



(a) Original (b) Noisy (c) Denoised Image

Fig. 9. *Example of non-spatially-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 12.77 dB, the resulting PSNR is 29.87 dB.*

similar and both achieve better results than those presented in the original inpainting algorithm [43].

C. Demosaicing

We now present results for demosaicing images. We ran our experiments on the images in Figure 12, 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

(a) Original Image (b) Damaged Image (c) Restored Image

Fig. 10. *From left to right : The 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 29.36 dB.*

(a) Original image (b) Image with text (c) Restored image

Fig. 11. *Inpainting for text removal.*

of size $5 \times 5 \times 3$, with a patch-sparsity factor $L = 20$, and compare to bilinear interpolation, the results given by Kimmel's algorithm [48], three very recent methods and state-of-the-art results presented in [46].

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 as well and sometimes better than state-of-the-art, improving by 0.11 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

Im	BI	K [48]	AP [47]	OR [49]	SA [52]	CC [46]	D1	D1L	D2	D2L
1	26.21	24.64	37.70	34.66	38.32	38.53	37.17	38.19	37.77	**38.97**
2	33.08	32.29	39.57	39.22	39.95	40.43	38.06	39.68	39.59	**40.81**
3	34.45	33.84	41.45	41.18	41.18	42.54	40.89	42.94	41.51	**43.19**
4	33.46	33.45	40.03	38.56	39.55	40.50	39.93	**41.46**	40.03	40.76
5	26.51	31.89	37.46	35.25	36.48	37.89	36.99	38.05	37.77	**38.24**
6	27.66	35.58	38.50	31.04	39.08	**40.03**	37.69	39.14	38.32	39.61
7	33.38	36.36	41.77	41.39	41.50	42.15	40.87	42.53	41.70	**42.79**
8	23.39	33.23	35.08	31.04	35.87	**36.41**	34.89	35.58	35.30	35.94
9	31.16	39.36	41.72	41.27	41.94	**43.04**	41.82	42.50	42.39	42.96
10	32.38	39.22	42.02	40.39	41.80	**42.51**	41.65	42.11	41.95	42.50
11	28.96	35.65	39.14	37.42	38.92	**39.86**	38.58	39.26	38.97	39.70
12	33.27	39.37	42.51	42.30	42.37	**43.45**	41.26	42.93	42.31	43.42
13	23.79	31.07	34.30	31.08	34.91	**34.90**	33.62	34.13	34.57	34.86
14	29.05	32.50	35.60	35.58	34.52	36.88	36.31	37.37	37.12	**37.87**
15	33.04	34.19	39.53	37.77	38.97	**39.78**	38.42	40.16	38.29	39.77
16	31.13	38.12	41.76	41.82	41.60	**43.64**	41.03	42.33	42.15	43.15
17	31.80	38.34	41.11	39.06	40.97	41.21	40.59	40.98	41.25	**41.72**
18	28.30	32.75	37.45	35.28	37.27	**37.49**	36.13	36.67	36.73	37.21
19	27.84	36.50	39.46	38.06	39.96	**41.00**	38.36	39.98	38.37	40.45
20	31.51	37.63	40.66	39.05	40.51	**41.07**	39.29	39.98	40.11	40.86
21	28.38	36.07	38.66	36.22	38.93	39.12	38.23	38.58	38.82	**39.27**
22	30.14	36.00	37.55	36.49	37.67	37.97	37.61	38.21	38.02	**38.56**
23	34.83	34.01	41.88	41.34	41.79	42.89	41.67	42.72	41.76	**42.96**
24	26.83	32.95	34.78	32.49	34.82	35.04	34.27	35.14	34.54	**35.44**
Avg.	30.06	35.21	39.15	37.70	39.12	39.93	38.55	39.61	39.14	**40.04**

TABLE III

Comparison of the PSNR, in dB, for different demosaicing algorithms on the Kodak data set. Some results are taken from the paper [46]. BI refers to a simple bilinear interpolation, then, K, AP, OR, SA, CC refer to the algorithms respectively from [48], [47], [49], [52], [46]. D1 refers to the results obtained with a globally trained dictionary (200 *iterations with* $L = 6$ *on* 200000 *different* $5 \times 5 \times 3$ *patches), D2 refers to a better trained dictionary (*600 *iterations with* $L = 6$ *on* 200000 *different* $5 \times 5 \times 3$ *patches). Then, D1L and D2L 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* 15*. Bold indicates the best results for each image, and these are mostly shared between our algorithm and the one recently reported in [46], which was explicitly designed for handling demosaicing problems.*

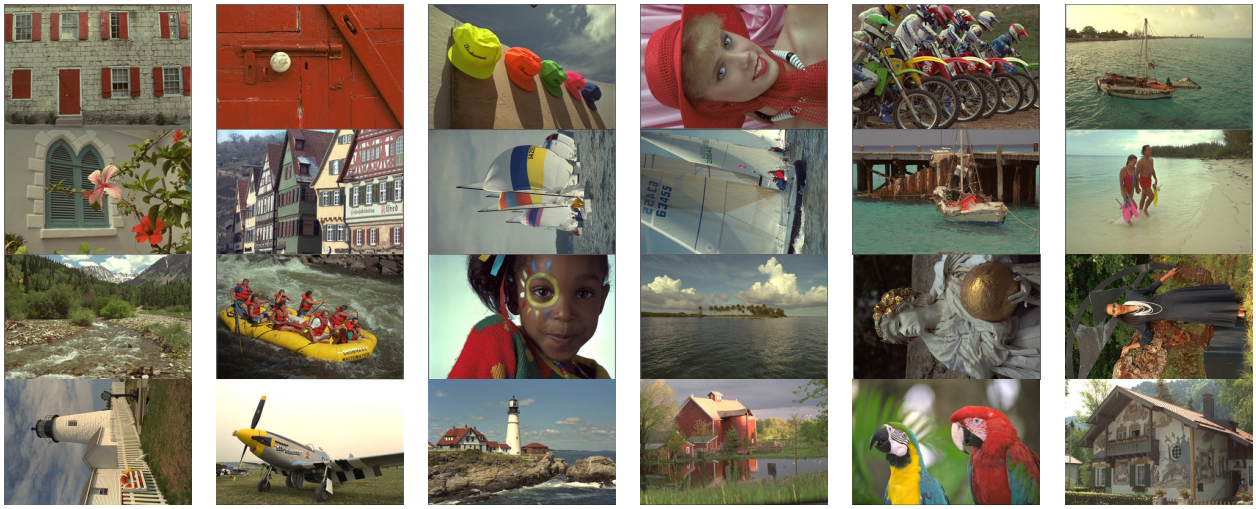


Fig. 12. *Kodak image database, images* 1 *to* 24 *from left to right and top to bottom, so that the upper left image is n1, and the upper right image is n6*

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 Figure 13.

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 gray-scale K-SVD algorithm introduced in [3], [2] 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.

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

(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

Fig. 13. *Examples of demosaicing results. The image 13(a) is one where we do not perform (visually) as well as the best interpolation-based algorithm. On the other ones, Figures 13(e), 13(g) and 13(c), we outperform these algorithms.*

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 gray-scale 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

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

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 [53]. The basic algorithm is presented on Figure 14. We observed some improvements with this two-scale dictionary, as presented in Table IV, encouraging our ongoing research on learning multiscale image models and multiscale sparsity frameworks.

Input: $\mathbf{y}$ (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^s_Γ.
- *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^b_Γ.
- *Computation of the optimal weights:* For each pixel (i, j) of each image m of Γ, denoted by Γ_{ijm}, compute the optimal weight:

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

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

Denoising process:

- *With small patches:* Denoise $\mathbf{y}$ using patches of size $s \times s$. This gives the reconstructed images $\hat{\mathbf{x}}^s$ and the sparsity of each pixel, L^s.
- *With large patches:* Denoise $\mathbf{y}$ using patches of size $b \times b$. This gives the reconstructed images $\hat{\mathbf{x}}^b$ and L^b.
- *Estimation of the optimal weights:* Use the Support Vector Regression to estimate the optimal weights for $\mathbf{x}$ 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 $\hat{\mathbf{x}}^s$ and $\hat{\mathbf{x}}^b$, with $\hat{\lambda}$ providing the weights.

Fig. 14. *K-SVD algorithm for denoising with two sizes of patches.*

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**

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.*

Concerning the computational complexity, the trade-off 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 important 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 gray-scale images 512×512, a one fourth reduction proved not to affect the quality of the reconstruction [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 multi-channels 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 Transactions on Signal Processing*, to appear.

[2] M. Elad and M. Aharon, "Image denoising via learned dictionaries and sparse representation," in *Proc. IEEE Computer Vision and Pattern Recognition (CVPR)*, New York, NY, USA, June 2006.

[3] ——, "Image denoising via sparse and redundant representations over learned dictionaries," *IEEE Transactions on Image Processing*, to appear.

[4] S. Mallat, *A Wavelet Tour of Signal Processing, Second Edition (Wavelet Analysis & Its Applications)*. Academic Press, September 1999.

[5] E. Candes and D. L. Donoho, "Recovering edges in ill-posed inverse problems: Optimality of curvelet frames," *Annals of statistics*, vol. 30, no. 3, pp. 784–842, June 2002.

[6] ——, "New tight frames of curvelets and the problem of approximating piecewise $\mathbf{C}^2$ images with piecewise $\mathbf{C}^2$ edges," *Comm. Pure Appl. Math.*, vol. 57, pp. 219–266, February 2004.

[7] M. Do and M. Vetterli, "Framing pyramids," *IEEE Transactions on Signal Processing*, vol. 51, no. 9, pp. 2329–2342, 2003.

[8] M. Do and M. Vertterli, *Contourlets, Beyond Wavelets*, G. V. Welland, Ed. Academic Press, 2003.

[9] D. Donoho, "Wedgelets: Nearly minimax estimation of edges," *Annals of statistics*, vol. 27, no. 3, pp. 859–897, June 1998.

[10] S. Mallat and E. L. Pennec, "Bandelet image approximation and compression," *SIAM Journal of Multiscale Modeling and Simulation*, to appear.

[11] ——, "Sparse geometric image representation with bandelets," *IEEE Transactions on Image Processing*, vol. 14, no. 4, pp. 423–438, 2005.

[12] W. T. Freeman and E. H. Adelson, "The design and the use of steerable filters," *IEEE Pat. Anal. Mach. Intell.*, vol. 13, no. 9, pp. 891–906, 1991.

[13] E. P. Simoncelli, W. T. Freeman, E. H. Adelson, and D. J. Heeger, "Shiftable multi-scale transforms," *IEEE Transactions on Informations Theory*, vol. 38, no. 2, pp. 587–607, September 1992.

[14] M. W. Marcellin, M. J. Gormish, A. Bilgin, and M. P. Boliek, "An overview of JPEG-2000," in *Data Compression Conference*, 2000, pp. 523–544.

[15] R. Eslami and H. Radha, "Translation-invariant contourlet transform and its application to image denoising," *IEEE Transactions on Image Processing*, 2004, to appear.

[16] B. Matalon, M. Elad, and M. Zibulevsky, "Improved denoising of images using modeling of the redundant contourlet transform," in *Proceedings of the SPIE conference wavelets*, vol. 5914, July 2005.

[17] J. Portilla, V. Strela, M. Wainwright, and E. P. Simoncelli, "Image denoising using scale mixtures of gaussians in the wavelet domain," *IEEE Transactions on Image Processing*, vol. 13, no. 4, pp. 496–508, 2004.

[18] J.-L. Starck, E. Candes, and D. L. Donoho, "The curvelet transform for image denoising," *IEEE Transactions on Image Processing*, vol. 11, no. 6, pp. 670–684, 2002.

[19] M. Elad, J.-L. Starck, P. Querre, and D. L. Donoho, "Simultaneous cartoon and texture image inpainting using morphological component analysis (mca)," *Journal on Applied and Computational Harmonic Analysis*, vol. 19, pp. 340–358, November 2005.

[20] G. M. Davis, S. Mallat, and Z. Zhang, "Adaptive time-frequency decompositions," *SPIE J. of Opt. Engin.*, vol. 33, no. 7, pp. 2183–2191, July 1994.

[21] S. Mallat and Z. Zhang, "Matching pursuit in a time-frequency dictionary," *IEEE Transactions on Signal Processing*, vol. 41, no. 12, pp. 3397–3415, 1993.

[22] 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*, November 1993.

[23] S. S. Chen, D. L. Donoho, and M. A. Saunders, "Atomic decomposition by basis pursuit," *SIAM Review*, vol. 43, no. 1, pp. 129–59, 2001.

[24] S. C. Zhu and D. Mumford, "Prior learning and gibbs reaction-diffusion," *IEEE Trans. on Pattern Analysis and Machine Intel.*, vol. 19, no. 11, pp. 1236–50, 1997.

[25] J. McAuley, T. Caetano, A. Smola, and M. O. Franz, "Learning high-order mrf priors of color images," *Proc. 23rd Intl. Conf. Machine Learning (ICML 2006)*, pp. 617–624, January 2006.

[26] S. Roth and M. J. Black, "Fields of experts: A framework for learning image priors." in *Proc. IEEE Computer Vision and Pattern Recognition (CVPR)*, San Diego, CA, USA, June 2005.

[27] R. W. Buccigrossi and E. P. Simoncelli, "Image compression via joint statistical characterization in the wavelet domain," *IEEE Transactions on Image Processing*, vol. 8, no. 12, pp. 1688–1701, 1999.

[28] E. Haber and L. Tenorio, "Learning regularization functionals," *Inverse Problems*, vol. 19, pp. 611–626, 2003.

[29] S. Baker and T. Kanade, "Limits on super-resolution and how to break them," *IEEE Trans. on Pattern Analysis and Maching Intel.*, vol. 24, no. 9, pp. 1167–1183, 2002.

[30] A. A. Efros and T. K. Leung, "Texture synthesis by non-parametric sampling," in *Proc. IEEE Int. Conference on Computer Vision (ICCV'99)*, Corfu, Greece, September 1999.

[31] W. T. Freeman, T. T. Jones, and E. C. Pasztor, "Example-based super-resolution," *IEEE Computer Graphics and Applications*, vol. 22, no. 2, pp. 56–65, 2002.

[32] E. C. P. W. T. Freeman and O. T. Carmichael, "Learning low-level vision," *Int. Journal of Computer Vision*, vol. 40, no. 1, pp. 25–47, 2000.

[33] A. Criminisi, P. Perez, and K. Toyama, "Region filling and object removal by exemplar-based image inpainting," *IEEE Transactions on Image Processing*, vol. 13, no. 9, pp. 1200–1212, 2004.

[34] D. Datsenko and M. Elad, "Example-based single document image super-resolution: A global map approach with outlier rejection," *Journal of Mathematical Signal Processing*, 2006, to appear.

[35] M. Elad and D. Datsenko, "Example-based regularization deployed to super-resolution reconstruction of a single image," *The Computer Journal*, 2006, to appear.

[36] M. F. Tappen, B. C. Russell, and W. T. Freeman, "Exploiting the sparse derivative prior for super-resolution and image demosaicing," in *In IEEE Workshop on Statistical and Computational Theories of Vision*, 2003.

[37] T. Weissman, E. Ordentlich, G. Seroussi, S. Verdu, and M. J. Weinberger, "Universal discrete denoising: known channel," *IEEE Transactions on Information Theory*, vol. 15, no. 1, pp. 5–28, 2005.

[38] A. Buades, B. Coll, and J.-M. Morel, "A non-local algorithm for image denoising," in *Proc. IEEE Computer Vision and Pattern Recognition (CVPR)*, San Diego, CA, USA, June 2005.

[39] A. Buades, B. Coll, and J. M. Morel, "A review of image denoising algorithms, with a new one," *Multiscale Modeling & Simulation*, vol. 4, no. 2, pp. 490–530, 2005.

[40] C. Kervrann and J. Boulanger, "Optimal spatial adaptation for patch-based image denoising," *IEEE Trans. on Image Processing*, vol. 15, no. 10, pp. 2866–2878, 2006.

[41] M. Mahmoudi and G. Sapiro, "Fast image and video denoising via non-local means of similar neighborhoods," *IEEE Signal Processing Letters*, vol. 2, no. 12, pp. 839–842, 2005.

[42] K. S. Miller, *Multidimensional Gaussian Distributions*. Wiley, 1964.

[43] M. Bertalmio, G. Sapiro, V. Caselles, and C. Ballester, "Image inpainting," in *SIGGRAPH '00: Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, 2000, pp. 417–424.

[44] N. Srebro and T. Jaakkola, "Weighted low-rank approximations." in *ICML*, 2003, pp. 720–727.

[45] A. Criminisi, P. Perez, and K. Toyama, "Object removal by exemplar-based inpainting," in *Proc. IEEE Computer Vision and Pattern Recognition (CVPR)*, Madison, WI, June 2003.

[46] K.-H. Chung and Y.-H. Chan, "Color demosaicing using variance of color differences," *IEEE Transactions on Image Processing*, vol. 15, no. 10, pp. 2944–2955, October 2006.

[47] B. K. Gunturk, Y. Altunbask, and R. M. Mersereau, "Color plane interpolation using alternating projections," *IEEE Transactions on Image Processing*, vol. 11, no. 9, pp. 997–1013, 2002.

[48] R. Kimmel, "Demosaicing: image reconstruction from color ccd samples." *IEEE Transactions on Image Processing*, vol. 8, no. 9, pp. 1221–1228, 1999.

[49] D. D. Muresan and T. W. Parks, "Demosaicing using optimal recovery," *IEEE Transactions on Image Processing*, vol. 14, no. 2, pp. 267–278, 2005.

[50] B. C. Russell, A. Torralba, K. P. Murphy, and W. T. Freeman, "Labelme: a database and web-based tool for image annotation," MIT, Tech. Rep., September 2005, aI Lab Memo AIM-2005-025.

[51] D. Martin, C. Fowlkes, D. Tal, and J. Malik, "A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics," in *Proc. 8th Int'l Conf. Computer Vision*, vol. 2, July 2001, pp. 416–423.

[52] X. Li, "Demosaicing by successive approximations," *IEEE Transactions on Image Processing*, vol. 14, no. 2, pp. 267–278, 2005.

[53] A. J. Smola and B. Schlkopf, "A tutorial on support vector regression," *Statistics and Computing*, vol. 14, no. 3, pp. 199–222, 2004.

[54] B. A. Olshausen and D. J. Field, "Sparse coding with an overcomplete basis set: a strategy employed by v1?" *Vision Res*, vol. 37, no. 23, pp. 3311–3325, December 1997.