Dictionary Learning for Deblurring and Digital Zoom

Florent Couzinie-Devy, Julien Mairal, Francis Bach, Jean Ponce

International Journal of Computer Vision manuscript No.
(will be inserted by the editor)

Dictionary Learning for Deblurring and Digital Zoom

Florent Couzinie-Devy · Julien Mairal · Francis Bach · Jean Ponce

Received: date / Accepted: date

Abstract This paper proposes a novel approach to image deblurring and digital zooming using sparse local models of image appearance. These models, where small image patches are represented as linear combinations of a few elements drawn from some large set (dictionary) of candidates, have proven well adapted to several image restoration tasks. A key to their success has been to learn dictionaries adapted to the reconstruction of small image patches. In contrast, recent works have proposed instead to learn dictionaries which are not only adapted to data reconstruction, but also tuned for a specific task. We introduce here such an approach to deblurring and digital zoom, using pairs of blurry/sharp (or low-/high-resolution) images for training, as well as an effective stochastic gradient algorithm for solving the corresponding optimization task. Although this learning problem is not convex, once the dictionaries have been learned, the sharp/high-resolution image can be recovered via convex optimization at test time. Experiments with synthetic and real data demonstrate the effectiveness of the proposed approach, leading to state-of-the-art performance for non-blind image deblurring and digital zoom.

Keywords deblurring · super-resolution · dictionary learning · sparse coding · digital zoom

F. Couzinie-Devy
Willow Project-Team, Laboratoire d'Informatique de l'École Normale Supérieure (ENS/INRIA/CNRS UMR 8548),
23, avenue d'Italie, CS 81321, 75214 Paris Cedex 13, France.
E-mail: couzinie@ens.fr

J. Mairal
Departement of Statistics, University of California, Berkeley
#301, Evans Hall, CA 94720-3860.
E-mail: julien@stat.berkeley.edu

F. Bach
Sierra Project-Team, Laboratoire d'Informatique de l'École Normale Supérieure (ENS/INRIA/CNRS UMR 8548),
23, avenue d'Italie, CS 81321, 75214 Paris Cedex 13, France.
E-mail: francis.bach@inria.fr

J. Ponce
Willow Project-Team, Laboratoire d'Informatique de l'École Normale Supérieure (ENS/INRIA/CNRS UMR 8548),
23, avenue d'Italie, CS 81321, 75214 Paris Cedex 13, France.
E-mail: jean.ponce@ens.fr

1 Introduction

With recent advances in sensor design, the quality of the signal output by digital reflex and hybrid/bridge cameras is remarkably high. Point-and-shoot cameras, however, remain susceptible to noise at high sensitivity settings and/or low-light conditions, and this problem is exacerbated for mobile phone cameras with their small lenses and sensor areas. Photographs taken with a long exposure time are less noisy but may be blurry due to movements in the scene or camera shake. Likewise, although the image resolution of modern cameras keeps on increasing, there is a clear demand for high-quality digital zooming from amateur and professional photographers, whether they crop their family vacation pictures or use footage from camera phones in newscasts. Thus, the classical image restoration problems of denoising, deblurring, multi-frame super-resolution and digital zooming (also called single-image super-resolution) are still of acute and in fact growing importance, and they have received renewed attention lately with the emergence of computational photography (e.g., [8,12, 16]).

The image deblurring problem is naturally ill-posed: Indeed, perfect low-pass filters remove all high-frequency information from images. They are non-invertible operators, and different sharp images can give rise to

the same blurry one. Thus, an appropriate image model is required to regularize the deblurring process. Several explicit priors for natural images have been proposed in the past for different tasks in image restoration. Early work relied on various smoothness assumptions, or image decompositions on fixed bases such as wavelets [20]. More recent approaches include non-local means filtering [1], learned sparse models [6,28,19], piecewise linear estimator [29], Gaussian scale mixtures [22], fields of experts [24], kernel regression [26], and block matching with 3D filtering (BM3D) [3]. Pairs of low-/high-resolution images have also been used as an *implicit* image prior in digital zooming tasks [11], and combining the exemplar-based approach with image self-similarities at different scales has recently led to impressive results [12].

We propose in this paper to build on several of these ideas with a new approach to non-blind image deblurring (the blur kernel is assumed to be fixed and known) and digital zooming. Like Freeman et al. [11], we use training pairs of blurry/sharp or low-/high-resolution image patches readily available for these tasks to learn our model parameters. We also exploit learned sparse local models of image appearance, as in [6,28], which have been known to be very effective for several image reconstruction tasks. Our method shares some ideas with the work of Yang et al. [28], but our formulation combines several novelties that improves the results:

- Whereas the approach of [28] is purely generative (this model learns how to simultaneously reconstruct pairs of low- and high-resolution patches), our approach learns how to reconstruct a high-resolution patch *given* a low-resolution one. In essence, the difference is the same as between generative and discriminative models in machine learning.

- We present a novel formulation for non-blind image deblurring and digital zooming, combining a *linear predictor* with *dictionary learning*, and show with extensive experiments on both synthetic and real data that our approach is competitive with the state of the art for these two tasks.

- We adapt the stochastic gradient descent of [17] for solving the corresponding learning problem allowing the use of large databases of training patches (typically several millions).

Notation. We define for $p \geq 1$ the ℓ_p norm of a vector $\boldsymbol{x}$ in $\mathbb{R}^m$ as $\|\boldsymbol{x}\|_p = (\sum_{i=1}^m |\boldsymbol{x}[i]|^p)^{1/p}$, where $\boldsymbol{x}[i]$ denotes the i-th coordinate of $\boldsymbol{x}$. We denote the Frobenius norm of a matrix $\mathbf{X}$ in $\mathbb{R}^{m\times n}$ by $\|\mathbf{X}\|_F = (\sum_{i=1}^m \sum_{j=1}^n |\mathbf{X}[i,j]|^2)^{1/2}$.

2 Related Work

2.1 Deblurring and Digital Zoom

Blur is a common image degradation, and the literature on the subject is quite large (see, e.g., [5,8–10,16,26]). Most existing methods assume a shift-invariant blur operator such that a blurry image $\mathbf{B}$ can be modelled as the convolution of the sharp image $\mathbf{S}$ with a fixed blur kernel $\mathbf{k}$:

$$\mathbf{B} = \mathbf{k} * \mathbf{S} + \mathbf{n}, \tag{1}$$

where $\mathbf{n}$ is an additive noise, usually i.i.d. Gaussian with zero mean. This model, while often satisfactory, does not take into account the fact that blur due to defocus or rotational camera motion is not uniform [16]. But, at least *locally*, it is sufficient to describe many types of blurs.

In the noiseless case when the filter is a known *imperfect* low-pass filter—that is, there is no zero in its Fourier transform, the blurring operator is invertible and deblurring amounts to inverting the Fourier transform. However, noise is always present in natural images, and even a small amount dominates the signal in high frequencies, leading to numerous artefacts. Regularization methods have been extensively studied to tackle this problem [14]. They usually impose smoothness constraints on the reconstructed images. The most recent and effective algorithms in this line of work usually adopt a two-step approach [4,10,13]: first, a simple regularized inversion of the blur is performed, then the resulting image is processed with classical denoising algorithms to remove artefacts. Various denoising methods have been used for this task: for instance, a Gaussian scale mixture model (GSM) [13], the shape-adaptive discrete cosine transform [10], or block matching with 3D-filtering kernel regression [4].

The digital zooming literature has seen in recent years the development of another line of research, following the exemplar-based method introduced by Freeman et al. [11]. Correspondences between high-resolution patches and low-resolution ones are learned by building a large database of such pairs. This idea has been successfully exploited by Glasner et al. [12], leading to state-of-the-art results. Along the same line, but using sparse image representations instead, pairs of corresponding patches are used by Yang et al. [28] to jointly learn high and low-resolution dictionaries. As shown in Section 3, the method we propose exploits these exemplar-based ideas as well, but in a significantly different way.

2.2 Learned Sparse Representations

Like several recent approaches to image restoration [6, 28], our method is based on the sparse decomposition of image patches. Using a dictionary matrix $\mathbf{D} = [\mathbf{d}_1, \ldots, \mathbf{d}_k]$ in $\mathbb{R}^{m \times k}$, a signal $\boldsymbol{x}$ in $\mathbb{R}^m$ is reconstructed as a linear combination of a few columns of $\mathbf{D}$, called atoms or dictionary elements. In typical image processing applications, m is relatively small, for instance $m = 64$ for image patches of size 8×8 pixels, and k can be larger than m, e.g., $k = 256$. We say that the dictionary $\mathbf{D}$ is well adapted to a vector $\boldsymbol{x}$ when there exists a sparse vector $\boldsymbol{\alpha}$ in $\mathbb{R}^k$ such that $\boldsymbol{x}$ can be approximated by the product $\mathbf{D}\boldsymbol{\alpha}$.

Exploiting these types of models usually requires a "good" dictionary. It can either be prespecified or designed by adapting its content to fit a given set of signal examples. Choosing prespecified atoms is appealing: The theoretical properties of the corresponding dictionaries can often be analysed, and, in many cases, it leads to fast algorithms for computing sparse representations. This is indeed the case for wavelets [20], curvelets, steerable wavelet filters, short-time Fourier transforms, etc. The success of the corresponding dictionaries in applications depends on how suitable they are to sparsely describe the relevant signals.

Another approach consists of learning the dictionary on a set of signal examples. The sparse decomposition of a patch $\boldsymbol{x}$ on a fixed dictionary $\mathbf{D}$ can be achieved by solving an optimization problem called Lasso in statistics [27] or basis pursuit in signal processing [2]:

$$\min_{\boldsymbol{\alpha} \in \mathbb{R}^k} \|\boldsymbol{x} - \mathbf{D}\boldsymbol{\alpha}\|_2^2 + \lambda \|\boldsymbol{\alpha}\|_1, \tag{2}$$

where the code $\boldsymbol{\alpha}$ in $\mathbb{R}^k$ is the representation of $\boldsymbol{x}$ over the dictionary $\mathbf{D}$, and λ is a parameter for controlling the sparsity of the solution.[1] Following an idea originally introduced in the neuroscience community by Olshausen and Field [21], Aharon et al. [6] have empirically shown that learning a dictionary $\mathbf{D}$ adapted to natural images could lead to better performance for image denoising than using off-the-shelf ones. For a database of n patches of size m, a dictionary is learned by solving the following optimization problem

$$\min_{\mathbf{D} \in \mathcal{D}, \boldsymbol{\alpha}_i \in \mathbb{R}^k} \frac{1}{n} \sum_{i=1}^{n} \|\boldsymbol{x}_i - \mathbf{D}\boldsymbol{\alpha}_i\|_2^2 + \lambda \|\boldsymbol{\alpha}_i\|_1, \tag{3}$$

where $\boldsymbol{x}_i$ is the i-th patch of the training set, and $\boldsymbol{\alpha}_i$ is its associated sparse code. To prevent the columns of $\mathbf{D}$ from being arbitrarily large (which would lead to arbitrarily small values of $\boldsymbol{\alpha}$), the dictionary $\mathbf{D}$ is constrained to belong to the set $\mathcal{D}$ of matrices in $\mathbb{R}^{m \times k}$ whose columns have an ℓ_2 norm less than or equal to one.

Several algorithms have been designed to address this problem. They either update $\mathbf{D}$ and the vectors $\boldsymbol{\alpha}_i$ in a sequential way [6], or are based on stochastic approximations [18,21].

2.3 Deblurring with Dictionaries

Several methods using dictionaries for deblurring have been presented in recent years [28,29]. Yu et al. [29], while not learning a dictionary as presented in the previous section, uses orthogonal basis obtained with principal component analysis (PCA). By "learning" several such dictionaries (one for each edge direction), and by choosing the best dictionary for each patch, the sharp patch can be reconstructed.

In the pioneering work by Yang et al. [28], a pair of dictionaries ($\mathbf{D}_b$,$\mathbf{D}_s$) is used, one dictionary for preprocessed blurred patches and the other for sharp patches. The preprocessing consists in the concatenation of oriented high-pass filters (gradients and Laplacian filters). During training, $\mathbf{D}_b$ and $\mathbf{D}_s$ are learned for representing simultaneously (with the same sparse code) the sharp patches with $\mathbf{D}_s$ and the preprocessed blurred patches with $\mathbf{D}_b$. At test time, given a new preprocessed blurry patch $\boldsymbol{x}$, a sparse code $\boldsymbol{\alpha}$ is obtained by decomposing $\boldsymbol{x}$ using $\mathbf{D}_b$, and ones hopes $\mathbf{D}_s\boldsymbol{\alpha}$ to be a good estimate of the unknown sharp patch.

This method, while appealing by its simplicity, suffers from an asymmetry between training and testing: Whereas in the learning phase, both blurred and sharp patches are used to obtain the sparse codes, at test time the code is only computed using the blurry patches. Our method addresses this problem by a different training formulation. Moreover preprocessing the data has empirically not shown to be necessary.

3 Proposed Approach

We show in this section how to learn dictionaries adapted to the deblurring and digital zoom tasks. As in exemplar-based methods [11,12,28], we are given a training set of n pairs of patches (obtained from pairs of blurry/sharp images), that are used to estimate model parameters. Unlike the classical dictionary learning problem of Eq. (3) which is unsupervised, our deblurring and digital zoom formulation is therefore supervised, trying to predict the sharp patches from the blurry ones.

[1] It is well known that ℓ_1 regularization yields a sparse solution for $\boldsymbol{\alpha}$, but there is no direct analytic link between the value of λ and the corresponding effective sparsity that it yields.

To predict a sharp pixel value, it is necessary to observe neighbouring blurry pixels. Sharp patches and blurry patches may therefore have different sizes, which we denote respectively by m_b and m_s, with m_b larger than m_s. During the test phase, we observe a test image $\mathbf{B}$ and try to estimate the underlying sharp image $\mathbf{S}$ according to Eq. (1), assuming of course that its blur is of the same nature as the one used during the training phase. The following sections present different formulations to recover an estimate of $\mathbf{S}$.

3.1 Linear Model

Blurring is, at least locally, a linear operation resulting from the convolution of a sharp image with a filter. When the support of the blur kernel is small compared to the patch sizes m_s and m_b, one can assume a linear relation between the blurry and sharp patches. Thus, a simple approach to the deblurring problem consists of learning how to invert this linear transform with a simple ridge regression model.

Training Step: A training set $(\mathbf{b}_i, \mathbf{s}_i)$, $i = 1, \dots, n$ of pairs of blurry/sharp patches is given. The training step amounts to finding the matrix $\mathbf{W}$ in $\mathbb{R}^{m_s \times m_b}$ that solves the following optimization problem:

$$\min_{\mathbf{W} \in \mathbb{R}^{m_s \times m_b}} \frac{1}{n} \sum_{i=1}^{n} \|\mathbf{s}_i - \mathbf{W}\mathbf{b}_i\|_2^2 + \mu \|\mathbf{W}\|_F^2, \tag{4}$$

where $\|\mathbf{W}\|_F$ denotes the Frobenius norm of the matrix $\mathbf{W}$, n is the number of training pairs of patches, and μ is a regularization parameter, which prevents overfitting on the training set and ensures that the learning problem is well posed. When n is very large (several millions), overfitting is unlikely to occur and setting μ to a small value (e.g., $\mu = 10^{-8}$ in our experiments) leads to acceptable results in practice. For this reason, and for simplifying the notation, we drop the term $\mu\|\mathbf{W}\|_F^2$ in the rest of the paper.

Testing Step: The parameters $\mathbf{W}$ are now fixed, and we are given a noisy test image $\mathbf{B}$, the goal being to recover a sharp estimate $\mathbf{S}$. However, as mentioned in Section 2, the noise dominates the signal in high frequencies, and in practice the linear model, which basically tries to invert the blur operator, leads to poor results despite the large amount of training data. Improvements can be achieved using recent denoising algorithms, either by pre-processing $\mathbf{B}$ to remove some of its noise, and/or by post-processing the sharp estimate to remove artefacts.

We now pre-process $\mathbf{B}$ and call $\tilde{\mathbf{B}}$ its denoised version, which is obtained with a denoising algorithm [19], and respectively denote by $\tilde{\mathbf{b}}_i$ and $\mathbf{s}_i$ the patches of $\tilde{\mathbf{B}}$ and $\mathbf{S}$ centered at the pixel i, using any indexing of the image pixels. Note that the patches $\mathbf{s}_i$ are here different from the ones in the training set, even though we use for simplicity the same notation. We assume with our learned linear model that the relation $\mathbf{s}_i \approx \mathbf{W}\tilde{\mathbf{b}}_i$ holds for the patch indexed by i. According to this model, the problem of reconstructing the sharp image $\mathbf{S}$ can be written as:

$$\min_{\mathbf{S}} \frac{1}{n_s} \sum_{i=1}^{n_s} \|\mathbf{s}_i - \mathbf{W}\tilde{\mathbf{b}}_i\|_2^2, \tag{5}$$

where n_s is the number of patches in the image $\mathbf{S}$. By using such a local linear model, and since the patches overlap, each pixel of the image $\mathbf{S}$ admits as many predictions as patches it belongs to. The solution of Eq. (5) is the average of the different predictions at each pixel, which is a classical way of aggregating estimates in patch-based methods [6].

This model is easy to optimize and to understand but has several limitations. First, small mistakes made during the denoising process can be amplified by the deblurring step.

Second, when the blur kernel totally suppresses some of the high frequencies of the image, putting them to zero, one cannot recover them with a local linear model: in the Fourier domain it correspond to a multiplication of the nullified coefficient by a finite number. This is one of the motivations for introducing a nonlinear model based on sparse representations to overcome these limitations.

3.2 Dictionary Learning Formulation

In a recent paper, Yang et al. [28] have shown that learning multiple dictionaries to establish correspondences between low- and high-resolution image patches is an effective approach to digital zoom. Following this idea, we propose to learn a pair of dictionaries $\mathbf{D}_s$ in $\mathbb{R}^{m_s \times k}$ and $\mathbf{D}_b$ in $\mathbb{R}^{m_b \times k}$ to reconstruct patterns that the linear model presented in the previous section cannot recover.

Training step: Given again a training set $(\mathbf{b}_i, \mathbf{s}_i)$, $i = 1, \dots, n$ of pairs of blurry-noisy/sharp patches, we address

$$\min_{\mathbf{D}_b \in \mathcal{D}, \mathbf{D}_s, \mathbf{W}} \frac{1}{n} \sum_{i=1}^{n} \|\mathbf{s}_i - \mathbf{W}\tilde{\mathbf{b}}_i - \mathbf{D}_s \boldsymbol{\alpha}^\star(\mathbf{b}_i, \mathbf{D}_b)\|_2^2, \tag{6}$$

where $\boldsymbol{\alpha}^\star(\mathbf{b}_i, \mathbf{D}_b)$ is the solution of the following sparse coding problem

$$\boldsymbol{\alpha}^\star(\mathbf{b}_i, \mathbf{D}_b) \triangleq \arg\min_{\boldsymbol{\alpha} \in \mathbb{R}^k} \|\mathbf{b}_i - \mathbf{D}_b\boldsymbol{\alpha}\|_2^2 + \lambda\|\boldsymbol{\alpha}\|_1, \tag{7}$$

which is unique and well defined under a few reasonable assumptions on the dictionary $\mathbf{D}_b$ (see [18] and references therein for more details).[2] The patch $\tilde{\mathbf{b}}_i$ is a denoisied version of $\mathbf{b}_i$. The matrices $\mathbf{D}_b$ and $\mathbf{D}_s$ are two dictionaries jointly learned such that for all i, $\mathbf{W}\tilde{\mathbf{b}}_i + \mathbf{D}_s\boldsymbol{\alpha}^\star(\mathbf{b}_i, \mathbf{D}_b)$ is a good estimator of the sharp patch $\mathbf{s}_i$. Summing two different predictors is a classical way of combining two models. In this case, we are hoping that the addition of the dictionary term to the linear term will permit a better recovery of the high frequencies. The two models are optimized jointly and are not just an averaging of two independent predictors.

Note that $\mathbf{D}_s$ does not need to be regularized in our formulation. We indeed assume that a large amount of training data is available and as a consequence our model does not suffer from overfitting.

Testing step: According to our model, and using the same notations as in Eq. (5), our estimate $\hat{\mathbf{S}}$ at test time is achieved by solving the following optimization problem

$$\min_{\mathbf{S}} \frac{1}{n_s} \sum_{i=1}^{n_s} \|\mathbf{s}_i - \mathbf{W}\tilde{\mathbf{b}}_i - \mathbf{D}_s\boldsymbol{\alpha}^\star(\mathbf{b}_i, \mathbf{D}_b)\|_2^2, \tag{8}$$

where $\mathbf{s}_i, \mathbf{b}_i, \tilde{\mathbf{b}}_i$ are respectively here the patches centered at pixel i of the sharp image $\mathbf{S}$, the blurry, noisy image $\mathbf{B}$ and the blurry, denoisied image $\tilde{\mathbf{B}}$.

The optimization problem defined in Eq. (6) is harder than the classical dictionary learning of Eq. (3) or the one formulated by Yang et al. [28], but this formulation presents advantages.

In the work of Yang et al. [28], the sparse coefficients $\boldsymbol{\alpha}$ are obtained during the training phase by jointly decomposing blurry patches $\mathbf{b}_i$ and sharp patches $\mathbf{s}_i$ onto two learned dictionaries $\mathbf{D}_b$ and $\mathbf{D}_s$. Such a model aims to ensure that there always exists a sparse code $\boldsymbol{\alpha}$ that both fits the patches $\mathbf{b}_i$ and $\mathbf{s}_i$. However, at test time, since the sharp patches are not available, the vectors $\boldsymbol{\alpha}$ can only be computed from blurry patches $\mathbf{b}_i$, and the fact that the resulting $\boldsymbol{\alpha}$ should be good for the corresponding sharp patch $\mathbf{s}_i$ is not guaranteed.

Our approach does not suffer from this issue since the sparse coefficients $\boldsymbol{\alpha}$ are always obtained on blurry patches only, both during the training and testing phase. We learn the dictionaries $\mathbf{D}_b$ and $\mathbf{D}_s$ and the linear predictor $\mathbf{W}$ such that $\mathbf{s}_i$ is well predicted *given* a patch $\mathbf{b}_i$. Whereas this solves the issue mentioned above, it leads to more challenging optimization problems than [28]. The optimization method we propose builds upon [17], which provides a general framework for solving such dictionary learning problems. The method is presented briefly in Section 4.

We have presented so far a framework adapted to the deblurring task, where we wanted to obtain a sharp image from a blurry one. The problem of digital zoom consists of increasing the resolution of an image, but can be formulated as a deblurring problem in a simple way: A low-resolution image can indeed be turned into a blurry high-resolution image with any interpolation technique, the task of digital zoom being then to *deblur* this new image. The training pairs of images can be generated by downsampling high-resolution images. Note that the antialiasing filter applied during downsampling and the choice of the interpolation method are important. We worked with the antialiasing from the Matlab function *imresize*.

4 Optimization

The formulation of Eq (6) for learning a pair of dictionaries $\mathbf{D}_b$ and $\mathbf{D}_s$ and a linear predictor $\mathbf{W}$ for the deblurring task is a large-scale learning problem, where many training samples $(\mathbf{b}_i, \mathbf{s}_i)$ can easily be available. The main difficulty in the optimization comes from the terms $\boldsymbol{\alpha}^\star(\mathbf{b}_i, \mathbf{D}_b)$, which are defined as solutions to the sparse coding problem of Eq. (7). The vectors $\boldsymbol{\alpha}^\star(\mathbf{b}_i, \mathbf{D}_b)$ therefore depend on the dictionary $\mathbf{D}_b$ and are not differentiable with respect to it, preventing us from using a direct gradient descent method.

However, despite these two drawbacks, it has been shown in [17] that such problems enjoy a few asymptotic properties that make it possible to use stochastic gradient descent when the number of training samples is large. Assuming an infinite training set $(\mathbf{b}_i, \mathbf{s}_i)$ that are i.i.d. samples drawn from some probability distribution, and under mild assumptions, we define the asymptotic cost function

$$\begin{aligned} f(\mathbf{D}_b, \mathbf{D}_s, \mathbf{W}) &\triangleq \lim_{n\to+\infty} \frac{1}{n}\sum_{i=1}^{n} \|\mathbf{s}_i - \mathbf{W}\mathbf{b}_i - \mathbf{D}_s\boldsymbol{\alpha}^\star(\mathbf{b}_i, \mathbf{D}_b)\|_2^2, \\ &= \mathbb{E}_{(\mathbf{b},\mathbf{s})}\big[\|\mathbf{s} - \mathbf{W}\mathbf{b} - \mathbf{D}_s\boldsymbol{\alpha}^\star(\mathbf{b}, \mathbf{D}_b)\|_2^2\big], \end{aligned} \tag{9}$$

where $(\mathbf{b}, \mathbf{s})$ are random variables distributed according to the joint probability distribution of low/high-resolution patches.

The optimization of cost functions that have the form of an expectation over a supposedly infinite training set is usually tackled with stochastic gradient techniques (see [17,18] and references therein), that are iterative procedures drawing randomly one element of the

[2] We have empirically found for our deblurring and super-resolution tasks on natural image patches and our dictionaries that the solution of Eq.(7) was always unique. For different tasks or data, the possible non-uniqueness of the Lasso solution could be an issue (see [17]).

training set at a time. Of course training sets are in practice finite, but we have empirically obtained good results by optimizing on a large training set of 10 millions of training patches. This is indeed the approach proposed in [17] for such problems, from which the following proposition can be derived.

Proposition 1 [Differentiability of f] *Assume that the training data $(\mathbf{b},\mathbf{s})$ admits a continuous probability density, and assume the same hypotheses on the dictionary $\mathbf{D}_b$ as in [17]. Then, f is differentiable and*

$$\begin{aligned}
\nabla_{\mathbf{W}} f &= -\mathbb{E}_{(\mathbf{b},\mathbf{s})}[2(\mathbf{s}-\mathbf{D}_s\boldsymbol{\alpha}^\star-\mathbf{W}\mathbf{b})\mathbf{b}^T],\\
\nabla_{\mathbf{D}_s} f &= -\mathbb{E}_{(\mathbf{b},\mathbf{s})}[2(\mathbf{s}-\mathbf{D}_s\boldsymbol{\alpha}^\star-\mathbf{W}\mathbf{b})\boldsymbol{\alpha}^{\star T}],\\
\nabla_{\mathbf{D}_b} f &= -\mathbb{E}_{(\mathbf{b},\mathbf{s})}[2(\mathbf{b}\boldsymbol{\beta}^{\star T}-\mathbf{D}_b\boldsymbol{\alpha}^\star\boldsymbol{\beta}^{\star T}-\mathbf{D}_b\boldsymbol{\beta}^\star\boldsymbol{\alpha}^{\star T})],
\end{aligned} \tag{10}$$

where $\boldsymbol{\alpha}^\star$ denotes $\boldsymbol{\alpha}^\star(\mathbf{b},\mathbf{D}_b)$, and

$$\boldsymbol{\beta}^\star_{\Lambda^C}=0 \text{ and } \boldsymbol{\beta}^\star_{\Lambda}=-(\mathbf{D}_{b\Lambda}^T\mathbf{D}_{b\Lambda})^{-1}\mathbf{D}_{s\Lambda}^T(\mathbf{s}-\mathbf{D}_s\boldsymbol{\alpha}^\star-\mathbf{W}\mathbf{b}), \tag{11}$$

where Λ denotes the indices of the nonzero coefficients of $\boldsymbol{\alpha}^\star$, for any vector $\mathbf{u}$, the vector $\mathbf{u}_\Lambda$ contains the values of the vector $\mathbf{u}$ corresponding to the indices Λ, and for any matrix $\mathbf{U}$, the matrix $\mathbf{U}_\Lambda$ contains the columns of $\mathbf{U}$ corresponding to the indices Λ.

Algorithm 1 presents our method for learning $\mathbf{D}_s, \mathbf{W}$ and $\mathbf{D}_b$. It is a stochastic gradient descent algorithm, which adapts [17] to our formulation. It draws randomly one element of the training set at each iteration, computes the terms inside the expectations of Eq. (10), and moves the parameters $\mathbf{D}_s, \mathbf{W}, \mathbf{D}_b$ one step in these directions.

Since $\mathbf{D}_b$ is constrained to be in the set $\mathcal{D}$ defined in Eq. (3), an orthogonal projection on this set is required at each iteration of the algorithm. It is denoted by $\Pi_{\mathcal{D}}$.

To improve the efficiency of the algorithm, we use a classical heuristic often referred to as : Instead of drawing a single pair of the training set at the same time, we draw η of them, e.g., $\eta = 500$, compute η directions given by Eq. (10), and move the model parameters $\mathbf{D}_b, \mathbf{D}_s, \mathbf{W}$ in the average direction. This improves the stability of the stochastic gradient descent algorithm, and experimentally gives a faster convergence. Since our optimization problem is not convex, it requires a good initialization. We proceed as follows: (i) We learn a dictionary $\mathbf{D}_b$ using the unsupervised formulation of Eq. (3) with the software[3] accompanying [18] on the set of patches $\mathbf{b}_i$. (ii) We fix $\mathbf{D}_b$ and optimize Eq. (6) with respect to $\mathbf{W}$ and $\mathbf{D}_s$, which is a convex optimization problem. In experiments, this procedure provides us with a good initialization.

[3] The SPAMS toolbox is an open-source software available at: *http://www.di.ens.fr/willow/SPAMS/*

Algorithm 1 Dictionary Learning for Deblurring and Digital Zoom

Require: $(\mathbf{b}_i,\mathbf{s}_i)$, $i=1,\ldots,n$ (training set); $\lambda,\mu\in\mathbb{R}$ (parameters); $\mathbf{D}_b\in\mathcal{D}$ (initial "blurry" dictionary), $\mathbf{D}_s$ (initial "sharp" dictionary); T (number of iterations); t_0,ρ (learning rate parameters for the stochastic gradient descent).
for $t=1$ to T **do**
  Draw $(\mathbf{b}_t,\mathbf{s}_t)$ from the training set.
  Sparse coding: compute $\boldsymbol{\alpha}^\star \triangleq \boldsymbol{\alpha}^\star(\mathbf{b}_t,\mathbf{D}_b)$.
  Compute the active set: $\Lambda \leftarrow \{j : \boldsymbol{\alpha}^\star[j]\neq 0\}$.
  Compute $\boldsymbol{\beta}^\star$ according to Eq. (11).
  Choose the learning rate $\rho_t \leftarrow \frac{\rho}{t+t_0}$.
  Update parameters:

$$\begin{aligned}
\mathbf{W} &\leftarrow \mathbf{W}+\rho_t(\mathbf{s}_t-\mathbf{D}_s\boldsymbol{\alpha}^\star-\mathbf{W}\mathbf{b}_t)\mathbf{b}_t^T,\\
\mathbf{D}_s &\leftarrow \mathbf{D}_s+\rho_t(\mathbf{s}_t-\mathbf{D}_s\boldsymbol{\alpha}^\star-\mathbf{W}\mathbf{b}_t)\boldsymbol{\alpha}^{\star T},\\
\mathbf{D}_b &\leftarrow \Pi_{\mathcal{D}}\Big[\mathbf{D}_b+\rho_t\big(\mathbf{b}\boldsymbol{\beta}^{\star T}-\mathbf{D}_b\boldsymbol{\alpha}^\star\boldsymbol{\beta}^{\star T}-\mathbf{D}_b\boldsymbol{\beta}^\star\boldsymbol{\alpha}^{\star T}\big)\Big].
\end{aligned}$$

end for
return $(\mathbf{D}_b,\mathbf{D}_s,\mathbf{W})$ (learned model parameters).

5 Experiments

We present here experimental results obtained with our method and comparisons with state-of-the-art methods. In all our experiments, after an initialization step described in the previous section, we use the stochastic gradient descent algorithm with one pass over a database of approximately 10 millions of training patches, which are extracted from a set of natural images. All the images from this dataset are unrelated with the images used for testing our method. Our implementation is coded in C++ and Matlab. Learning a dictionary takes usually a few hours on a recent computer, while testing an image is faster (less than one minute for most of our test images).

5.1 Non-Blind Deblurring with Isotropic Kernels

To compare our method for the non-blind deblurring task, we have chosen a classical set of images and types of blurs, which has been used in several recent image processing papers (see [29] and references therein). Even though addressing such a synthetic non-blind deblurring task of course slightly deviates from real restoration problems with digital cameras, it is still an active topic in the image processing community, and has in fact proven useful in the past, leading to high-impact

Table 1 Experiments settings for the non-blind deblurring.

Exp.	Blur kernel $\mathbf{k}$	noise σ^2
1	9 x 9 uniform blur	0.308
2	$\mathbf{k}(x_1, x_2) = 1/(1 + x_1^2 + x_2^2)$	2
3	$\mathbf{k}(x_1, x_2) = 1/(1 + x_1^2 + x_2^2)$	8
4	$\mathbf{k} = [1\ 4\ 6\ 4\ 1]^T[1\ 4\ 6\ 4\ 1]/256$	49
5	Gaussian blur of variance $\sigma_b = 1$	25
6	Gaussian blur of variance $\sigma_b = 2$	25

applications in astronomic imaging [25] for example (see Section 5.2).

The different combinations of blurs and noises are detailed in Table 5.1, with the shape of the blur kernel and the variance of the noise (which is Gaussian and white). They are used in other papers and go from strong-blur/weak-noise to weak-blur/strong-noise cases.

For each blur level, we have generated pairs of blurry/sharp images from our training database, and learned dictionaries of size $k = 512$ elements. We have observed that the results quality usually improves with the dictionary size, 512 being a good compromise between quality and computational cost. Since our database is large, the parameters μ is always set to a negligible value, $\mu = 10^{-8}$. The size of patches m_s and m_b are respectively set to 7 and 11 for all experiments. The only parameter which should be carefully tuned to obtain good results is the regularization parameter λ. Following [4,10,13], we have manually chosen a value of λ via a rough grid search for each type of blur and used it for every image. We report quantitative results in Table 5.1 in terms of improvement in signal-to-noise ratio (ISNR),[4] and compare our method to the classical Richardson-Lucy algorithm [23], and to recent state-of-the-art methods [4,10, 16]. A few values are missing in the table: these experiments were not done by the authors of the papers. We observe that our method is competitive or better than the state of the art in experiments $2, 3, 4, 5, 6$, where the supports of the blur kernels are relatively small. On the contrary, our algorithm is significantly behind other approaches in the case 1, probably because our patches are too small compared to the kernel size. The simple linear model while not at the state of the art, is giving surprisingly good results for most of the blurs. Its combination with the dictionaries shows a significant improvement, leading to state-of-the-art performances. Qualitative results are presented in Figures 1, 2 and 3.

[4] Denoting by MSE the mean-squared-error for images whose intensities are between 0 and 255, the PSNR is defined as PSNR $= 10\log_{10}(255^2/\text{MSE})$ and is measured in dB. A gain of 1dB reduces the MSE by approximately 20%.

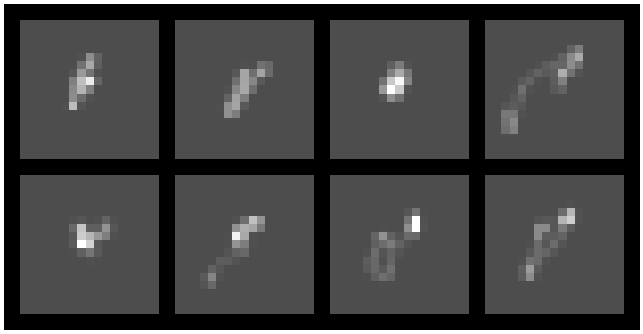

Fig. 5 Anisotropic kernels from [16] used in our experiments.

5.2 Astronomical Images

Our method is not designed specifically for the restoration of natural images. It adapts itself to the training set and can so be applied on various data. This versatility is illustrated here on astronomical imaging, which is a field where non-blind deblurring has had a major industrial impact. The experiment setting is based on a classical astronomical case. A star image has to be recovered from a blurred and noisy version of it. The blur kernel is the Hubble Space Telescope kernel as given in [25]. The additive noise is Gaussian. The training set is constructed from several others star images.

Figure 4 presents the results with several deblurring algorithm. Our method result is quantitatively better than the other algorithms: While the two algorithms adapted to natural images [4,15] gives a PSNR of 30.8 and 31.3, our method gets 33.5. In particular, our algorithm manages to recover really high values on the brightest stars. This is not surprising, several of these algorithms use priors that do not fit well astronomical images, but it validates the capability of our method to adapt to various data.

5.3 Non-Blind Deblurring with Anisotropic Kernels

While deblurring isotropic blurs is sufficient in many applications, anisotropic blurring appears in practical cases, e.g., camera-shaking blur. To test our algorithm on this setting, we used the kernels from the database by Levin et al. [16]. The local nature of our algorithm makes computationally challenging the treatment of large blurs and so we only worked with downsampled versions of the proposed kernels (by a factor 2). The 8 kernels used are shown in Figure 5. White Gaussian noise of variance 2 is added to the blurry images before deblurring. We compare in Table 3 with the sparse-gradient-based algorithm from Levin et al. [15] which is, to the best of our knowledge, the one giving the state-of-the-art results for this type of kernels.

Table 2 Isotropic deblurring results in ISNR (PSNR improvement). For each image/experiment, the best result is in bold. Four values are missing: the results for this experiment were taken from [29], who does not test on the exact same set of images than us.

	Exp. 1	Exp. 2	Exp. 3	Exp. 4	Exp. 5	Exp .6	Exp. 1	Exp. 2	Exp. 3	Exp. 4	Exp. 5	Exp .6
	Cameraman						*Lena*					
PSNR input image	20.76	22.35	22.29	24.7	25.53	23.44	25.84	27.57	27.35	29.00	30.74	28.97
Richardson-Lucy [23]	4.47	5.53	3.58	0.49	1.21	1.04	4.80	5.29	2.71	0.02	0.26	0.53
Sparse gradient [15]	7.73	6.89	4.78	2.24	2.64	2.70	7.02	2.83	5.44	4.06	3.30	3.33
SA-DCT [10]	**8.55**	8.11	6.33	3.37	-	-	7.79	7.55	6.10	4.49	3.56	3.46
BM3D [4]	8.34	8.19	6.40	3.34	3.73	**3.83**	**7.97**	**7.95**	**6.53**	4.81	4.18	4.12
Linear	3.34	7.72	6.00	3.20	3.47	2.69	3.58	7.30	5.82	4.64	3.89	3.58
Linear + Dictionary	4.76	**8.35**	**6.47**	**3.57**	**3.94**	3.35	4.83	7.79	6.13	**5.16**	**4.34**	**4.17**
	House						*Barbara*					
PSNR input image	24.11	26.28	26.10	28.51	30.16	28.18	22.49	23.49	23.35	24.28	25.02	23.46
Richardson-Lucy [23]	6.46	5.86	3.68	0.04	0.25	0.59	2.26	2.70	1.13	-0.06	0.12	0.02
Sparse gradient [15]	10.16	8.03	6.43	4.09	3.47	3.92	2.88	6.87	1.51	0.57	0.66	1.11
SA-DCT [10]	10.5	9.02	7.74	4.99	4.14	4.21	4.79	5.45	2.54	1.31	-	-
Dabov et al. [4]	**10.85**	**9.32**	**8.14**	5.13	4.79	**5.30**	**5.86**	**7.80**	3.94	1.90	**3.17**	**1.94**
Linear	4.25	8.90	7.58	5.22	4.51	4.26	2.39	7.18	4.27	1.86	2.89	1.56
Linear + Dictionary	6.99	**9.32**	7.71	**5.74**	**4.98**	5.09	2.65	7.64	**4.59**	**2.00**	3.11	1.70

Table 3 Anisotropic deblurring results in mean ISNR (PSNR improvement) over 5 images. The kernels used are downsampled versions of those from [16].

Kernel	1	2	3	4
Sparse gradient [15]	9.04	6.91	7.49	**10.67**
Ours	**10.67**	**7.17**	**9.02**	6.63
Kernel	5	6	7	8
Sparse gradient [15]	8.64	9.18	**11.15**	**10.24**
Ours	**10.52**	**10.03**	9.64	7.75

Our method does significantly worse than [15] on three of these kernels: there are the ones where the kernel is large and we think it is probably due to the locality of our predictor. For the 8 kernels, we worked with patches of size 13 and it might be not sufficient for too big kernels.

5.4 Digital Zoom

Following the same experimental protocol than for the deblurring experiments, we have evaluated our method for the digital zooming task. The dictionary size is $k = 512$, and the patch sizes are $m_b = 11$ and $m_s = 7$. Digital zooming is usually done on good quality images, with a very small noise: for this reason we use a small regularization parameter λ, which is set to 0.005.

It is always difficult to evaluate quantitatively the results of digital zoom algorithms. Indeed, upsampling and downsampling methods are often subject to subpixel misalignments, which are visually imperceptible, but make important mean square error differences. Moreover, the antialiasing filter that has to be applied during the downsampling is rarely detailed, making comparisons difficult. For this experiment, we used the Matlab function *imresize* with a bicubic interpolation to create the low-resolution images. The choice of the antialiasing, which allows to create the training set, is really important. With a too strong antialiasing our method might sharpen too much the images, while with a weak antialiasing it might not deblur enough.

We compare quantitatively with the method from Yang et al. [28] that also uses dictionaries, proving the efficiency of the discriminative approach. The dictionaries sizes are the same as ours (512), and the parameter λ is chosen on a validation set of images. This method works in two steps, first, it predicts a high-resolution image from a filtered version of the low resolution one using pairs of dictionaries, then, the image is cleaned using a backprojection. We compare the results at both steps with our method in Table 4.

Our method outperforms the full method from Yang et al. [28] by a small margin. But their results obtained only with dictionaries are significatively worse than ours. The discriminative learning of the dictionaries and the addition of the linear predictor improve greatly the results. Figure 6 compares our results with the ones of Yang et al. using one image from [28]. We have observed that both methods improve significantly upon the bicubic interpolation and gives similar results (with the backprojection step for Yang et al. [28] method).

We have also compared qualitatively our method with others works: In Figure 7, we present digital zooming results (by a factor 4) obtained on one image from [7,12]. Our results are in general slightly better visually than [7] (see the texture of the baby's hat for instance),

Fig. 1 Examples of deblurring and close-up for the case 2. First two lines, from top to bottom, left to right: original image, blurry image, Richardson-Lucy, sparse gradient [15], SA-DCT [10], our method. Last two lines: close-ups in the same order. Best seen by zooming on a computer screen.

but slightly behind [12] in terms of sharpness of edges (e.g. the baby's mouth). On the other hand, Glasdner et al. [12]'s algorithm reconstructs sometimes structures not present in the original image (e.g., square edges in the baby eye). In textured areas, we perform as good as [12].

6 Conclusion

In this paper, we have presented a new formulation for image deblurring and digital zooming using a supervised formulation of dictionary learning combined with a linear predictor. With a stochastic gradient descent algorithm, our approach is efficient and allows the use of millions of training samples. Experiments on natural

Fig. 2 Examples of deblurring for the case 3. First two line, from top to bottom, left to right: original image, blurry image, Richardson-Lucy, sparse gradient [15], SA-DCT [10], our method. Last two lines: close-ups in the same order. Best seen by zooming on a computer screen.

images show that our method is competitive with the state of the art for the non-blind deblurring and digital zooming tasks. Future work will consist of extending the approach to the blind deblurring problem, where a blur kernel has to be learned at the same time as the learned dictionaries, and exploiting self-similarities in images, which have proven to be very successful for digital zooming [12] and image denoising[19].

Acknowledgements This research was partially supported by the Agence Nationale de la Recherche (MGA Project) and the European Research Council (SIERRA and VideoWorld projects). In addition, Julien Mairal has been supported in part by the NSF grant SES-0835531 and NSF award CCF-0939370. The authors would like to thanks Jean-Luc Starck for sharing the astronomical data used in subsection 5.2 and Jianchao Yang for providing us with his code of digital zooming.

Fig. 3 Examples of deblurring for the case 4. From top to bottom, left to right: original image, blurry image, Richardson-Lucy, sparse gradient [15], SA-DCT [10], our method. Best seen by zooming on a computer screen.

Table 4 Digital zoom (by a factor 2) quantitative results in PSNR. We present two values for Yang et al. method: the first one is the result given by their dictionaries, the second one is obtained by adding a backprojection algorithm to the dictionaries. For each image, the best result is in bold.

	Cubic spline	Yang et al. [28]	Ours
Lena	31.91	32.13 / 33.06	**33.31**
Girl	31.44	31.48 / 31.93	**32.00**
Flower	38.48	38.69 / 39.59	**39.92**

References

1. Buades, A., Coll, B., Morel, J.: A non-local algorithm for image denoising. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2005)
2. Chen, S., Donoho, D., Saunders, M.: Atomic decomposition by basis pursuit. SIAM Journal on Scientific Computing **20**, 33–61 (1999)
3. Dabov, K., Foi, A., Katkovnik, V., Egiazarian, K.: Image Denoising by Sparse 3-D Transform-Domain Collabora-

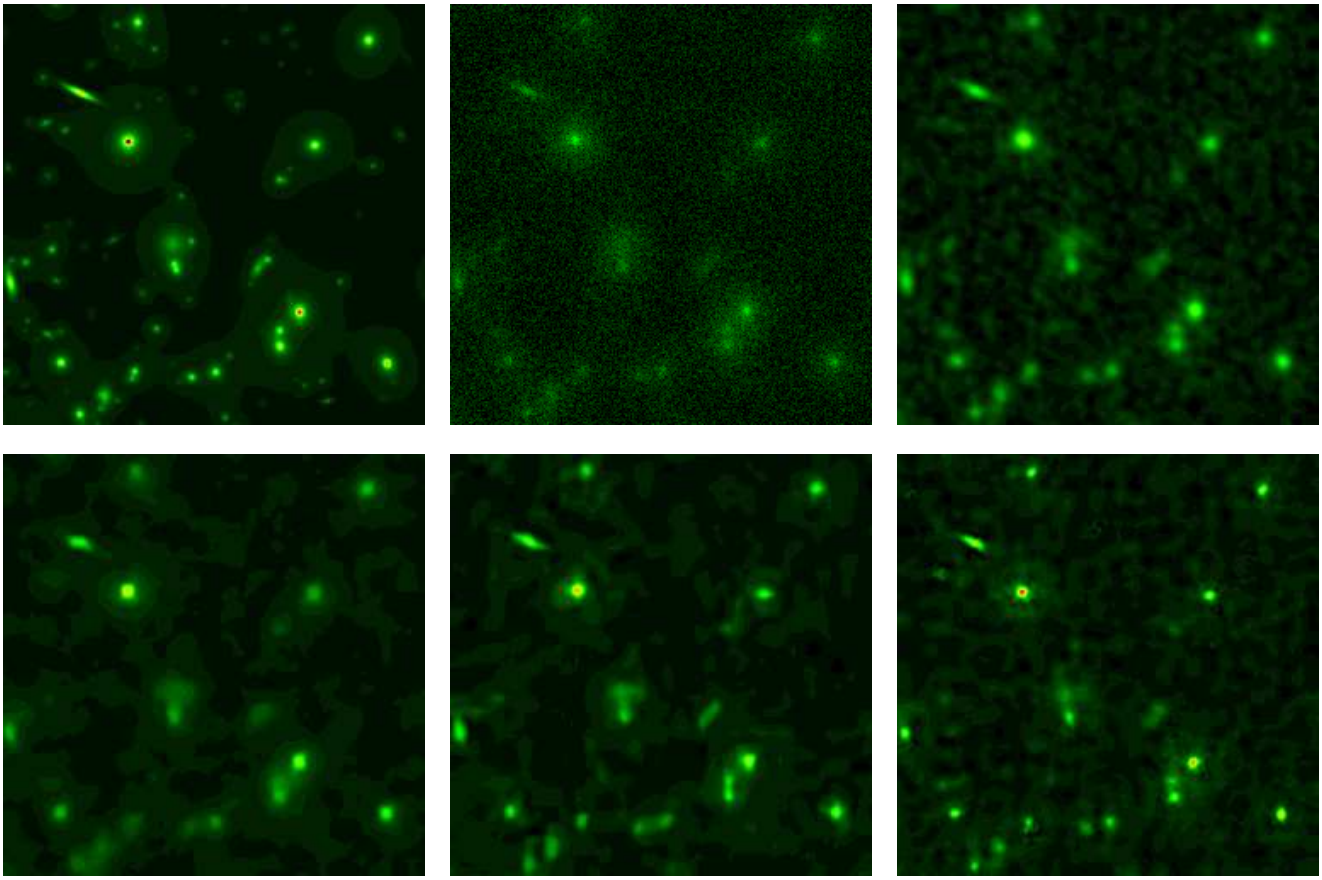

Fig. 4 Example of deblurring of an astronomical image. Between parenthesis is indicated the PSNR. First line: Original, blurred and noisy image (25.3) , Wiener deblurring (29.6). Second line: sparse gradient [15] (31.3), BM3D [4] (30.8), our method (33.5). Best seen in color.

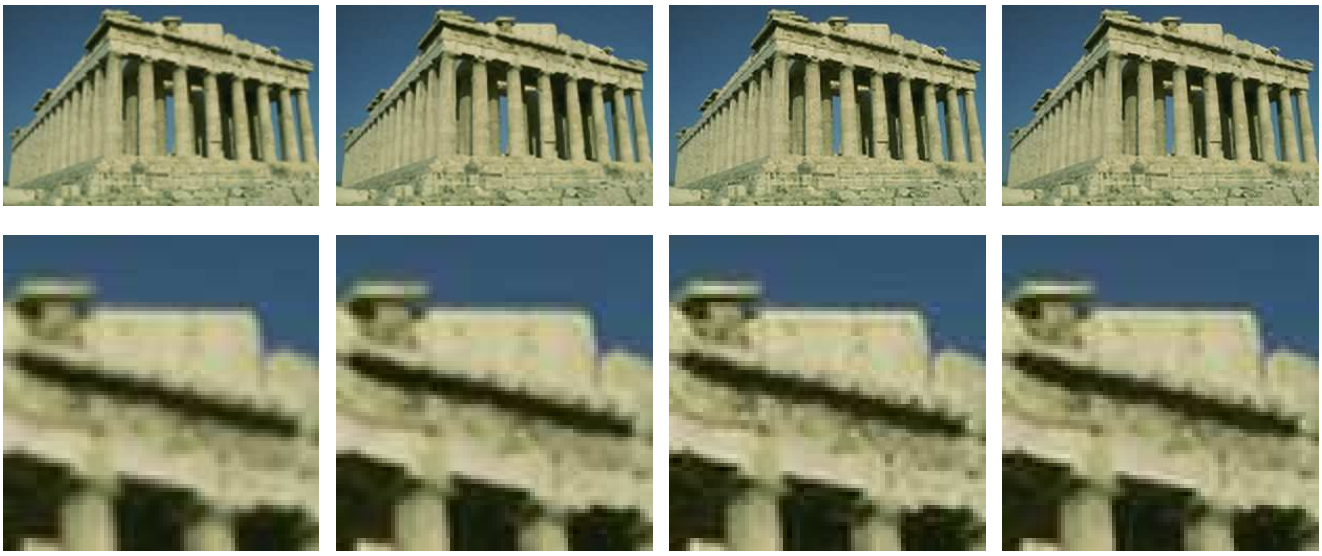

Fig. 6 Digital zoom by a factor 2. The top line shows the full image and the bottom one a zoom on one section. From left to right: bicubic interpolation, Yang et al. [28] (dictionary only), Yang et al. [28] with backprojection, our results

Fig. 7 Digital zoom by a factor 4. From left to right: bicubic interpolation, Fattal et al [7], Glasner et al. [12], our results. Best seen by zooming on a computer screen.

tive Filtering. IEEE Transactions on Image Processing **16**(8), 2080–2095 (2007)
4. Dabov, K., Foi, A., Katkovnik, V., Egiazarian, K.: Image restoration by sparse 3D transform-domain collaborative filtering. In: SPIE Electronic Imaging, vol. 6812 (2008)
5. Dias, J.: Fast gem wavelet-based image deconvolution algorithm. IEEE International Conference on Image Processing (2003)
6. Elad, M., Aharon, M.: Image denoising via sparse and redundant representations over learned dictionaries. IEEE Transactions on Image Processing **54**(12), 3736–3745 (2006)
7. Fattal, R.: Image upsampling via imposed edge statistics. ACM Transactions on Graphics **26**(3) (2007)
8. Fergus, R., Singh, B., Hertzmann, A., Roweis, S.T., Freeman, W.T.: Removing camera shake from a single photograph. ACM Trans. Graph. **25**(3), 787–794 (2006)
9. Figueiredo, M., Nowak, R.: A bound optimization approach to wavelet-based image deconvolution. IEEE International Conference on Image Processing (2005)
10. Foi, A., Dabov, K., Katkovnik, V., Egiazarian, K.: Shape-adaptive DCT for denoising and image reconstruction. In: Proceedings of SPIE, vol. 6064, pp. 203–214 (2006)
11. Freeman, W., Jones, T., Pasztor, E.: Example-based super-resolution. IEEE Computer Graphics and Applications pp. 56–65 (2002)
12. Glasner, D., Bagon, S., Irani, M.: Super-resolution from a single image. In: Proceedings of the IEEE International Conference on Computer Vision (ICCV) (2009)
13. Guerrero-Colon, J., Mancera, L., Portilla, J.: Image restoration using space-variant gaussian scale mixtures in overcomplete pyramids. IEEE Transactions on Image Processing **17**(1), 27–41 (2008)
14. Hansen, P.: Rank-deficient and discrete ill-posed problems: numerical aspects of linear inversion. Society for Industrial Mathematics (1998)
15. Levin, A., Fergus, R., Durand, F., Freeman, W.: Deconvolution using natural image priors. ACM Transactions on Graphics **26**
16. Levin, A., Weiss, Y., Durand, F., Freeman, W.: Understanding and evaluating blind deconvolution algorithms. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2009)
17. Mairal, J., Bach, F., Ponce, J.: Task-Driven Dictionary Learning. IEEE Transactions on Pattern Analysis and Machine Intelligence (2011). To appear.

18. Mairal, J., Bach, F., Ponce, J., Sapiro, G.: Online learning for matrix factorization and sparse coding. Journal of Machine Learning Research **11**, 19–60 (2010)
19. Mairal, J., Bach, F., Ponce, J., Sapiro, G., Zisserman, A.: Non-local sparse models for image restoration. In: Proceedings of the IEEE International Conference on Computer Vision (ICCV) (2009)
20. Mallat, S.: A Wavelet Tour of Signal Processing, Second Edition. Academic Press, New York (1999)
21. Olshausen, B., Field, D.: Sparse coding with an overcomplete basis set: A strategy employed by v1? Vision research **37**(23), 3311–3325 (1997)
22. Portilla, J., Strela, V., Wainwright, M., Simoncelli, E.: Image denoising using scale mixtures of Gaussians in the wavelet domain. IEEE Transactions on Image Processing **12**(11), 1338–1351 (2003)
23. Richardson, W.: Bayesian-based iterative method of image restoration. Journal of the Optical Society of America **62**(1), 55–59 (1972)
24. Roth, S., Black, M.J.: Fields of experts: A framework for learning image priors. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2005)
25. Starck, J., Murtagh, F.: Astronomical image and data analysis. Springer-Verlag (2006)
26. Takeda, H., Farsiu, S., Milanfar, P.: Deblurring using regularized locally adaptive kernel regression. IEEE Transactions on Image Processing **17**(4), 550–563 (2008)
27. Tibshirani, R.: Regression shrinkage and selection via the lasso. J. Royal. Statist. Soc B. **58**(1), 267–288 (1996)
28. Yang, J., Wright, J., Huang, T., Ma, Y.: Image superresolution via sparse representation. IEEE Transactions on Image Processing **19**(11), 2861–2873 (2010)
29. Yu, G., Sapiro, G., Mallat, S.: Solving Inverse Problems with Piecewise Linear Estimators: From Gaussian Mixture Models to Structured Sparsity. Preprint arXiv:1006.3056 (2010)