# Blind and Semi-Blind Deblurring of Natural Images

Mariana S. C. Almeida and Luís B. Almeida

***Abstract*—A method for blind image deblurring is presented. The method only makes weak assumptions about the blurring filter and is able to undo a wide variety of blurring degradations. To overcome the ill-posedness of the blind image deblurring problem, the method includes a learning technique which initially focuses on the main edges of the image and gradually takes details into account. A new image prior, which includes a new edge detector, is used. The method is able to handle unconstrained blurs, but also allows the use of constraints or of prior information on the blurring filter, as well as the use of filters defined in a parametric manner. Furthermore, it works in both single-frame and multiframe scenarios. The use of constrained blur models appropriate to the problem at hand, and/or of multiframe scenarios, generally improves the deblurring results. Tests performed on monochrome and color images, with various synthetic and real-life degradations, without and with noise, in single-frame and multiframe scenarios, showed good results, both in subjective terms and in terms of the increase of signal to noise ratio (ISNR) measure. In comparisons with other state of the art methods, our method yields better results, and shows to be applicable to a much wider range of blurs.**

***Index Terms*—Blind image deconvolution, image deblurring, image enhancement, image restoration, sparse distributions.**

## I. Introduction

IMAGE deblurring is an inverse problem whose aim is to recover an image from a version of that image which has suffered a linear degradation, with or without noise. This blurring degradation can be shift-variant or shift-invariant. Although there have been some proposed methods for recovering shift-variant linear degradations [1]–[8], the majority of existing deblurring methods was developed for invariant degradations, and the blind recovery from shift-invariant degradations is still considered a rather challenging problem. This paper focuses on shift-invariant blurs, and, in the context of this paper, "blur" will refer to a linear, shift-invariant degradation, i.e., a convolution, with or without noise, unless stated otherwise.

Automatic image deblurring is an objective of great practical interest for the enhancement of images in photo and video cameras [9]–[11], in astronomy [12], in remote sensing [13], in tomography [14], [15], in other biomedical imaging techniques [16]–[18], etc.

Image deblurring methods can be divided into two classes: *nonblind*, in which we assume the blurring operator to be known, and *blind*, in which we assume that the blurring operator is unknown. The method that we describe here belongs to the latter class. The application range of nonblind methods is much narrower than the one of blind methods: in most situations of practical interest the blurring filter's impulse response, also called point spread function (PSF), is not known with good accuracy. Since nonblind deblurring methods are very sensitive to mismatches between the PSF used by the method and the true blurring PSF, a poor knowledge of the blurring PSF normally leads to poor deblurring results.

Despite its narrower applicability, nonblind deblurring already is a difficult problem. The main difficulty faced by nonblind deblurring methods has to do with the presence of noise in the blurred image. Since the blurring operator typically is very ill-conditioned, this noise, even if very weak, can strongly contaminate the deblurred image. The problem is serious in situations in which the blurring PSF is exactly known, and gets worse if there is even a slight mismatch between the PSF used for deblurring and the one that caused the blur. Most nonblind deblurring methods [19]–[23] overcome this difficulty through the use of prior information about the image to be recovered, often doing this within a Bayesian or maximum *a posteriori* framework.

In blind image deblurring (BID), not only the degradation operator is ill-conditioned, but the problem also is, inherently, severely ill-posed: there is an infinite number of solutions (original image + blurring filter) that are compatible with the degraded image. For an overview of BID methods, see [24] and [25].

Most previously published blind deblurring methods are very limited, since they do not allow the use of a generic PSF. Most of them are based, instead, on PSF models with a small number of parameters [26]–[30]. For example, to model an out-of-focus blur, they normally use a circle with uniform intensity, having as single parameter the circle's radius [26]. Similarly, to model a motion blur, they normally use a straight-line segment with uniform intensity, the only parameters being length and slope [26]–[28]. These approaches are very limited, because such models rarely fit actual blurring PSFs well. For example, the out-of-focus blurring PSF generally is more complex than a simple uniform circle, and the camera motion that causes a motion blur generally is much more complex than a uniform, straight-line motion. And, as was emphasized above, even a slight mismatch between the deblurring PSF and the blurring PSF strongly degrades the quality of the deblurred image.

A recent work [30] manages to estimate the blur under the variational Bayesian approach. However, this method models the blur by means of a Gaussian filter, which is completely de-

Manuscript received July 21, 2008; revised July 27, 2009. First published August 28, 2009; current version published December 16, 2009. This work was supported in part by the Portuguese FCT and by the "Programa Operacional Sociedade do Conhecimento (POS-Conhecimento)", comparticipated by the FEDER European Community fund, under project POSC/EEA-CPS/61271/2004 and under the Ph.D. fellowship SFRH/BD/23919/2005. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Luminita Vese.

The authors are with the Instituto de Telecomunicações, Instituto Superior Técnico, 1049-001 Lisboa, Portugal (e-mail: mariana.almeida@lx.it.pt; lbalmeida@lx.it.pt).

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

Digital Object Identifier 10.1109/TIP.2009.2031231

fined by a single parameter (the Gaussian's variance), and is a very weak model for real-life blurs.

In an attempt to encompass less restrictive blurs, a fuzzy technique that uses several prespecified PSF models has been considered in [31]. Another blind deconvolution method, which is fast and has a proof of convergence, is described in [32]. However, this method assumes that the PSF is zero-phase and, furthermore, depends on the existence of a good initial estimate of the PSF.

References [33] and [34] present a method called APEX. Although this method covers some blurs which can be found in real-life, it is limited to blurring PSFs modeled by a symmetrical Lévy distribution with just two parameters. In Section IV-C, we present an experimental comparison of our method with APEX.

Some methods have been proposed, which impose no strong restrictions on the blurring filter [2], [35]–[38]. These methods typically impose priors over the blurring filter, and do not seem to be able to handle a wide variety of blurs and scenes. In [37] and [2], total variation (TV) is used to regularize the blurring filters. Besides being used for space-invariant blurs, the method described in [2] was also applied with success in a synthetic image with a space-variant blur. We present an experimental comparison with that method in Section IV-C. The method recently presented in [9] is much less restrictive than parameterized ones and yields good results, but is only designed for motion blurs.

An interesting method for blind deblurring of color images was proposed in [39]. This method appears not to pose any strong restrictions on the blurring filter. In the cited paper, several experimental results on synthetic blurs are shown, but little information is provided about them. From the information that is given, it appears that the blurring filters that were used in the experiments were either circularly symmetric (including simulated out-of-focus blurs), or corresponded to straight-line motion blurs. There seems to be no reason for the method not to be able to successfully deal with other kinds of blurs, however. The blurring PSFs that are shown in that paper appear to have a maximum size of about $5 \times 5$ pixels (or a length of 3 pixels, in the case of the motion blur). The improvements in signal to noise ratio (ISNR, see Section III for a definition) seem to be between 2 and 4 dB for the circularly symmetric blurs, and of 7 dB for the motion blur. The experimental results presented in Section IV show that, with much stronger blurs (much larger blurring PSFs), our method normally yielded larger improvements than the method of that paper.

In some cases, one has access to more than one degraded image from the same original scene, a fact which can be used to reduce the ill-posedness of the problem [6], [40]–[44]. There are also solutions like the ones presented in [45]–[47], which cannot be considered completely blind, since they require the use of additional data for preliminary training.

Contrary to previously published blind deconvolution methods such as those mentioned above, the method that we propose only makes a weak assumption on the blurring PSF: it must have a limited support. The method also assumes that the leading (most important) edges of the original image, before the blur, are sharp and sparse, as happens in most natural images. To the authors' knowledge, this is the first method to be proposed, which is able to yield results of good quality in such a wide range of situations.

The method uses a new prior which depends on the image's edges, and which favors images with sparse edges. This prior leads to a regularizing term which generalizes the well known total variation (TV), in its discrete form [48]. The estimation is guided to a good solution by first concentrating on the main edges of the image, and progressively dealing with smaller and/or fainter details. Though the method allows the use of a very generic PSF, it can also take into account prior information on the blurring PSF, if available. If a parameterized model of the PSF is known, the method allows the estimation of the model's parameters. Although initially developed for the single-frame scenario, the method can also be used in multiframe cases, benefiting from the existence of the additional information from the multiple frames.

The performance and the robustness of the method were tested in various experiments, with synthetic and real-life degradations, without and with constraints on the blurring filter, without and with noise, using monochrome and color images, and under the single-frame the multiframe paradigms. The quality of the results was evaluated both visually and in terms of ISNR. Detailed comparisons with two other methods available in the literature [2], [33] were performed, and show that the proposed method yields significantly better results than these other methods.

This paper is organized as follows: Section II describes the proposed method and presents the new prior for images. Section III describes some special aspects of the computation of the ISNR measure in the blind deblurring case. Experimental results are presented in Section IV. Section V presents conclusions and future research directions.

## II. Deblurring Method

The degradation that we aim to recover from, is modeled by

$$y = h * x + n \tag{1}$$

in which $y$, $x$ and $n$ are images which represent, respectively, the degraded image, the original image and additive noise; $h$ is the PSF of the blurring operator, and $*$ denotes the mathematical operation of convolution.

The deblurring method is based on two simple facts.

- In a natural image, leading edges are sparse.
- Edges of a blurred image are less sparse than those of a sharp image because they occupy a wider area.

Due to these facts, a prior which tends to make the detected edges sparser will tend to make the image sharper, while preventing it from becoming unnatural (i.e., from presenting noise or artifacts).

Let us designate by $f(\cdot)$ an edge detection operation, such that $f(x)$ is an image with the intensities of the edges that exist in the image $x$. The deblurring method that we propose finds a local minimum, with respect to both the image $x$ and the blur $h$, of the cost function

$$C(x,h) = \frac{1}{2}\|y - h * x\|_2^2 + \lambda R\left[f(x)\right] \tag{2}$$

TABLE I
DEBLURRING METHOD

| |
|---|
| *Initialization:* |
| 1 – Set $h$ to the identity operator. |
| 2 – Set $x$ equal to $y$. |
| 3 – Set $\lambda$ and $q$ to the initial values of the respective sequences. |
| *Optimization loop:* |
| 4 – Find a new $x$ estimate: $x = \text{argmin}_x\ C(x,h)$ |
| 5 – Find a new $h$ estimate: $h = \text{argmin}_h\ C(x,h)$ |
| 6 – Set $\lambda$ and $q$ to the next values in sequence. |
| 7 – If $\lambda \geq \lambda_{min}$ go back to 4; otherwise stop. |

where $R[f(x)] = R_f(x)$ is a regularizing term which favors solutions in which the edges present in $f(x)$ are sparse, and $\lambda$ is a regularization parameter. More details on the edge detector $f(\cdot)$ are given in Section II-A. Both the regularizer $R_f(.)$ and the prior from which it is obtained are described in Section II-B. The estimate of the blurring filter's PSF, $h$, is restricted to have a limited support, which should not be smaller (but may be larger) than the support of the actual blur.

A local minimum of the cost function, that corresponds to a good deblurring result, is reached by starting with a large value of the regularizing parameter $\lambda$ and progressively reducing it. That local minimum usually is not the global one. For example, for images with a significant amount of texture, we have found that the (image + filter) pair formed by the blurred image and the identity filter often yields a lower value of the cost function than the value that we obtain at the end of the deblurring process, when the estimated image is much sharper than the blurred one. We do not know whether the global minimum of the cost function would yield a good deblurred image, but we have reasons to believe that it might not do so. Of course, this leads us to think that there should be a better cost function, whose global minimum will yield a good deblurred image. We do not presently know such a function, however, and this clearly is an important topic for further research.

The deblurring method is outlined in Table I, for which a decreasing sequence of values of $\lambda$ and a nonincreasing sequence of values of $q$ are assumed to have been previously chosen ($q$ controls the regularizer's sparsity, as will be discussed ahead). In our experiments, we have used a geometric progression for the sequence of values of $\lambda$ ($\lambda_{n+1} = \lambda_n / r$). The filter estimation, performed in step 5 of the method, can take into account constraints or prior information on the filter, if these are available.

In the early stages of the deblurring process, the estimate of $h$ is still poor, and a strong regularization is required to make edges sharp in the estimated image, and to eliminate the wrong high frequency components that could otherwise appear. During these early iterations, $\lambda$ is large and only the main edges of the estimated image survive (see an example in Fig. 1). The surviving edges are sharp, however, due to the strong edge-sparsifying regularization, and these sharp edges allow the estimate of the blurring filter to be improved. In the next iteration, the improved filter, together with a somewhat lower value of $\lambda$, allows some smaller and/or fainter edges to be extracted. As the iteration proceeds and the filter estimate improves, smaller and/or fainter features are progressively handled, at a rate which is controlled by the rate of decrease of the value of $\lambda$. This procedure

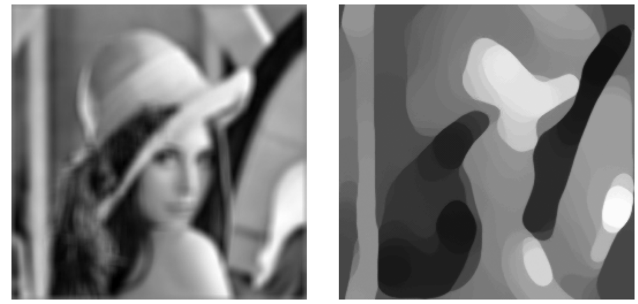

Fig. 1. Left: "Lena" image blurred with a 9 × 9 square blur. Right: Image estimate obtained for the first value of $\lambda$.

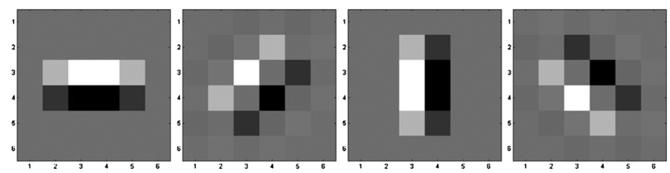

Fig. 2. Set of edge detection filters, in the four orientations that were used. The leftmost filter is the base filter $d_0$.

results in a guidance of the optimization, which leads it to a local minimum that normally corresponds to a good deblurring result.

## A. Edge Detector

In order to be able to apply a prior over the image edges, an edge detector was developed. This edge detector showed to yield better deblurring results than those obtained using detectors described in the literature, such as [49]–[52]. The edge detector is based on a set of filters obtained, from a base filter $d_0$, through successive rotations (see Fig. 2). The base filter (leftmost filter in Fig. 2) is a detector of edges in a given direction and, naturally, its rotated versions are detectors of edges in the corresponding rotated directions. Designating by $d_\theta$ the filters of the set and by

$$g_\theta(x) = d_\theta * x \tag{3}$$

the filters' outputs, the output of the edge detector is given by a combination of those outputs through an $L_2$ norm

$$f(x) = \sqrt{\sum_{\theta \in \Theta} g_\theta(x)^2} \tag{4}$$

in which $\Theta$ is the set of filter orientations under consideration. As an example of the detector's operation, Fig. 3 shows the edges that were extracted, from the "Lena" image, by this edge detector, using the set of filters shown in Fig. 2.

## B. Image Prior

The prior that we use for images assumes that edges are sparse, and that edge intensities at different pixels are independent from one another (which obviously is a large simplification, but still leads to good results). The edge intensity at each pixel $i$, denoted $f_i(x)$, is assumed to follow a sparse prior with density

$$p\left[f_i(x)\right] \propto e^{-k[f_i(x)+\epsilon]^q} \tag{5}$$

Fig. 3. Edges of "Lena" computed using the proposed edge detector.

where $k$ adjusts for the scale of edge intensities and $q$ controls the prior's sparsity; $\epsilon$ is a small parameter which allows us to obtain finite lateral derivatives at $f = 0$ (with $0 < q < 1$), making the prior closer to actual observed distributions, and also making the optimization easier.

Assuming, for the noise $n$ in (1), a Gaussian prior with zero mean and variance $\sigma^2$, the likelihood of the estimated pair (image + filter) is given by

$$p(x,h|y) \propto e^{-\frac{1}{2\sigma^2}\|y-h*x\|_2^2} \prod_i e^{-k[f_i(x)+\epsilon]^q} \tag{6}$$

where $i$ is an index running through all pixels. The log-likelihood is, apart from an irrelevant constant

$$L(x,h|y) = -\frac{1}{2\sigma^2}\|y-h*x\|_2^2 - k\sum_i ([f_i(x)+\epsilon]^q\ . \tag{7}$$

Maximizing this likelihood is equivalent to minimizing the cost function

$$C(x,h) = \frac{1}{2}\|y-h*x\|_2^2 + \lambda\sum_i ([f_i(x)+\epsilon]^q \tag{8}$$

where $\lambda = k\sigma^2$. This cost function is of the form given in (2). We can identify the data fidelity term, $(1/2)\|y-h*x\|_2^2$, and the regularizer, $R[f(x)] = \sum_i [f_i(x)+\epsilon]^q$. The regularizer $R(f_i)$ is plotted in Fig. 4, for the parameters that were used in the experiments, and for typical values of the edge intensity.

This regularizer was chosen both because it favors sharp edges and because, for certain values of the parameters, the corresponding prior is rather close to actual observed distributions of the edges obtained from our edge extractor. A regularizer which favors sparse edges, such as this one, which is nonsmooth and nonconvex, typically favors piece-wise constant image estimates [53]. In our method this is quite visible in the first iterations (see Fig. 1), but is almost imperceptible in the final estimate, if there is no noise, because the final regularization is then very weak. When the image to be deblurred is noisy, the final regularization cannot be made so weak, to prevent the appearance of a strong amount of noise in the deblurred image, and that image will still retain some piece-wise constant character, as can be seen in the experimental results shown in Section IV. This is a compromise that has to be made, in our method and in several other ones, when one is simultaneously doing deblurring and denoising.

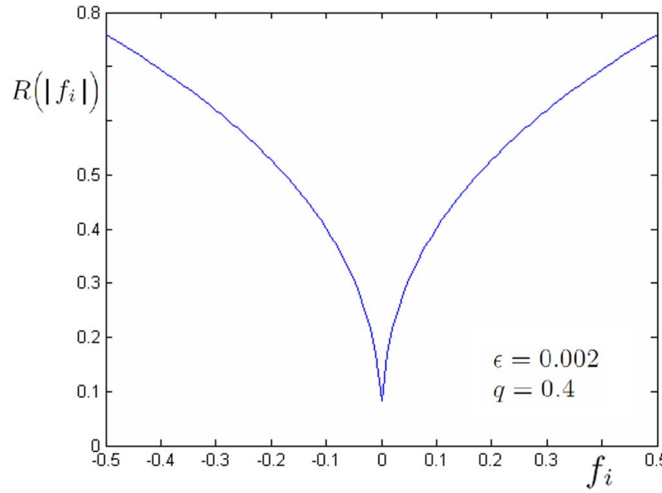


Fig. 4. Plot of $R(|f_i|)$, computed for the values of the parameters used in the experimental tests.

The cost function of (8) is similar to cost functions that have been used in other works on image deblurring (see [37] and [38], for example). The well known total variation regularizer (in its discrete form) is a special case of our regularizer: it is obtained by using just two filters which compute horizontal and vertical differences, and by setting $q = 1$ and $\epsilon = 0$. Despite these similarities with other methods, there are some important differences, in our approach, that are worth emphasizing.

- We use more elaborate edge detection filters than the simple horizontal and vertical differences used in many other works.
- We use $q < 1$, which makes the cost function nonconvex and, therefore, harder to optimize, but yields considerably better results.
- We use an optimization technique that leads to a good local minimum. As noted above, it is possible that the global minimum would not yield good deblurring results, but that is not the minimum that we seek in our method.

These three differences are crucial for the performance achieved by our method.

As was said above, we decrease $\lambda$ during the optimization. Therefore, except for the last phase of the optimization, $\lambda$ is not given by $k\sigma^2$. And in fact, even during that last phase, $\lambda$ still is not, in general, given by that expression, because the noise $n$, besides allowing for possible noise in the blurred image, also allows for a mismatch between the estimated filter $h$ and the true one. This mismatch leads to a difference between the reconstructed blurred image and the observed one, this difference being treated by the method as noise.

### C. Border Effects

Since, in the first iterations of the method, the image estimates only contain the main image edges, the optimal filter estimates, at this stage, would not (even approximately) have a limited support. We constrain, in step 5 of the method, the filter estimate to have a limited support. This gives rise to undesired border effects in that estimate [see Fig. 5(a)]. These effects decrease in subsequent iterations, as the estimate of the deblurred image

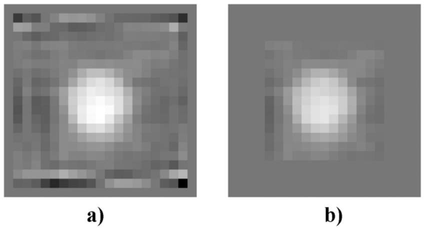

Fig. 5. Filter estimate at an early stage of the method: (a) with the safety zone; (b) after discarding the safety zone.

gets better. To avoid the influence of these effects, we use, in step 5 of the method, a "safety zone" with a width of a few pixels, around the desired filter support, and discard this zone, in each iteration, after the filter has been estimated [see Fig. 5(b)].

A border effect of a different kind, this time relating to the estimated image, is due to the fact that, near the border of the image, a part of the filter's support would fall outside the image. There is, therefore, a zone, adjacent to the image border, where estimation cannot be correctly performed. Estimation in this zone would typically lead to ringing artifacts parallel to the image borders. This border zone is not estimated, and is not included in the cost function of (8).

### D. Color and Hyperspectral Images

The method that we propose can also address color and hyperspectral images. A color image is a multichannel image which typically has three channels (red, green and blue). A hyperspectral image takes this concept much further, usually containing more than one hundred channels, which correspond to different frequency bands. In what follows we'll speak of color images, but what is said can be extended, without change, to hyperspectral images.

To restore a color image, each of its channels should be restored. However, during the restoration process, one should take into account that the various channels should remain aligned with one another. In the case of color images, even a small misalignment between the channels would lead to a significant visual degradation (color fringing along edges). Consequently, the color channels should be jointly processed, so that they maintain their alignment. A simple way to favor aligned solutions, is to apply the regularizer to the sum of the edge images from the three channels, instead of applying it separately to each channel. This corresponds to using the regularizer

$$R_f(x) = R[f(x)] \tag{9}$$

$$= R\left[\sum_c f^c(x)\right] \tag{10}$$

$$= \sum_i \left[\sum_c f_i^c(x) + \epsilon\right]^q \tag{11}$$

in which $f^c$ is the edge image computed by applying $f(.)$ to the $c^{th}$ color channel of $x$, and $f_i^c(x)$ is the $i$th pixel of that image. The image

$$f(x) = \sum_c f^c(x) \tag{12}$$

is the edge image obtained, from the color image $x$, by separately applying the edge extractor $f(.)$ to each of the channels, and adding the results.

If we assume that all channels have suffered the same blurring degradation, we only need to estimate a single blurring filter. In this case, the cost function which is used to recover the color image is given by

$$C(x,h) = \frac{1}{2}\sum_c \|y_c - h * x_c\|_2^2 + \lambda \sum_i [f_i(x) + \epsilon]^q \tag{13}$$

in which $x_c$ is the $c^{th}$ channel of the estimated image $x$, $y_c$ is the $c^{th}$ channel of the degraded image $y$ and $f_i(x)$ is, as before, the $i$th pixel of the image $f(x)$. On the other hand, if we assume that different channels have suffered different blurs (which can happen, for example, if there is a significant amount of chromatic aberration from the lens that produced the image), then the cost function should be

$$C(x,h) = \frac{1}{2}\sum_c \|y_c - h_c * x_c\|_2^2 + \lambda \sum_i [f_i(x) + \epsilon]^q \tag{14}$$

where $h_c$ is the blur corresponding to the $c^{th}$ channel.

### E. Multiframe Scenarios

The method can also be easily extended to address multiframe scenarios, in which one has several frames, each with its own degradation, but all obtained from the same sharp scene. In this case, we can take advantage of the extra information that results from the existence of more than one blurred image of the same scene. Instead of using a single degradation model, the data connection term of the cost function must now take into account the degradation models of the various acquired frames. We index the frames with the subscript $s$, and assume that each acquired frame, $y_s$, was degraded by a different blurring operator, $h_s$, and a different additive noise, $n_s$

$$y_s = h_s * x + n_s. \tag{15}$$

Assuming that all frames have Gaussian noises with the same variance, the multiframe cost function is

$$C(x,h) = \frac{1}{2}\sum_s \|y_s - h_s * x\|_2^2 + \lambda R_f(x). \tag{16}$$

### F. Filter Prior

The method was developed so that it would yield a good restoration performance without using any "strong" information on the blurring filter. Nevertheless, if prior information about the blurring filter is available, it can be used to advantage. If hard constraints on the blurring filter are known, they can be used in the filter optimization step (step 5 in Table I). "Soft" constraints on the filter can be incorporated through the use of an additional regularizing term. If we assume a prior over the blurring filter

$$p(h) \propto e^{k_h R_h(h)} \tag{17}$$

we are led to the cost function

$$C(x,h) = \frac{1}{2}\|y - h * x\|_2^2 + \lambda R\left[f(x)\right] + \lambda_h R_h(h) \tag{18}$$

in which $\lambda_h$ and $R_h(h)$ are, respectively, the regularizing parameter and the regularizing term of the blurring filter. $R_h(h)$ can be a TV regularizer, as in [2] and [37], for example, but other regularizers can also be used.

## III. Quality Measure

The measure that we used for evaluating the quality of the results of blind deblurring tests was the *increase in signal to noise ratio* (ISNR), similarly to what is commonly done in nonblind deblurring. However, the computation of a meaningful ISNR in blind deblurring situations raises some special issues that we now address.

We start by recalling the basic concept of ISNR. Assume that $x_0$ is an original image, $y$ is a degraded version of that image and $x$ is a recovered (enhanced) image, obtained from $y$. We start by defining the "signal" as image $x_0$, the "noise" of $y$ as $y - x_0$, and the "noise" of $x$ as $x - x_0$. The ISNR of the recovered image $x$ relative to the degraded image $y$ is, then, the difference between the SNR of $x$ and the SNR of $y$. It can be computed, in decibels, as

$$\mathrm{ISNR} = 10 \log_{10} \frac{\sum_i \left(y^i - x_0^i\right)^2}{\sum_i \left(x^i - x_0^i\right)^2} \tag{19}$$

where the superscript $i$ indexes the images' pixels, and the sums run through all pixels.

The special issues that arise in the computation of this measure in blind deblurring situations, are due to the following. The blind deblurring problem is strongly ill-posed. This means that nonregularized solutions have a large variability. There are two different kinds of variability that we need to distinguish here. One corresponds to changes in the *shape* of the estimated blurring filter's PSF $h$, compensated by matching changes in the estimated image $x$. In this case, different estimated images will, in general, exhibit different amounts of residual blur and/or different artifacts (e.g., ringing), which affect their quality. These degradations should be taken into account by the quality measure. However, two forms of variability that are of a different kind are 1) affine transformations of the intensity scale of the filter, compensated by affine transformations of the estimated image, and 2) small translations of the blurring filter's PSF, compensated by opposite translations of the estimated image. These degradations do not affect the quality of the deblurred image, and the restoration measure should be insensitive to them. As an example of the latter kind of variability, filter translations are visible in the positions of some of the estimated filters obtained in our tests, which are not fully centered (see Figs. 11 and 12).

The translation variability gets a bit more involved when we take into account that the images are processed in discrete form: There can be translations by a fractional number of pixels, which do not correspond to simple discrete translations of the discrete images, and involve interpolation between pixels.

To address these invariance issues, we have performed an image adjustment (spatial alignment and intensity rescaling) before comparing the images with the original sharp one. The estimated image was spatially aligned, and the pixel intensities were rescaled by an affine transformation, so as to minimize the image's squared error relative to the original sharp image. As a result, the noise energy of an image $x$ relative to the sharp image $x_0$ was given by

$$N(x) = \min\nolimits_{a,b,\Delta_h,\Delta_v} \|a x_{\Delta_h,\Delta_v} + b - x_0\|_2^2 \tag{20}$$

in which $x_{\Delta_h,\Delta_v}$ is the $x$ image shifted by $\Delta_h$ and $\Delta_v$ pixels in the horizontal and vertical directions, respectively, and $a$ and $b$ are the parameters of the affine transformation.

The spatial alignment was performed with a resolution of 1/4 pixel, and with a maximum shift of 3 pixels in each direction. The estimated image $x$ was saturated to the maximum and minimum values of the degraded image $x_0$, before alignment and rescaling. For the comparison to be fair, the alignment and rescaling were performed on both the deblurred image and the blurred one.

The ISNR of the recovered image $x$ relative to the blurred image $y$ was then computed as

$$ISNR = 10 \log_{10} \frac{N(y)}{N(x)}. \tag{21}$$

The sum on $i$, involved in the computation of $N(\cdot)$ [see (20)] was restricted to the valid pixels. By "valid pixels" we mean all pixels of the image, except for the zone, adjacent to the image borders, where the estimation could not be correctly performed, as explained in Section II-C, this zone being augmented by a width of 3 pixels to account for the maximal possible displacement due to the spatial alignment. The Matlab routines for image adjustment (alignment and rescaling) and for computing the ISNR are available at http://www.lx.it.pt/mscla/BID_QM.htm.

In order to be able to compare the reconstruction of the color images with the reconstruction of the corresponding grayscale ones, the ISNR of color images was computed on the luminance component $I(x)$ through the expression used in the NTSC television standard, obtained from the image's RGB channels $x_r$, $x_g$, and $x_b$ through

$$I(x) = 0.2989 x_r + 0.5870 x_g + 0.1140 x_b. \tag{22}$$

## IV. Experimental Results

We tested the proposed method both on synthetic blurs and on actual blurred photos. We also performed comparisons with two other blind deblurring methods, published in the literature [2], [33].

Our method was implemented as outlined in Table I, with the edge detection filters that are shown in Fig. 2. The PSF of the base filter that was used to generate these edge detection filters (i.e., the PSF of the leftmost filter in Fig. 2) was

$$d_0 = \begin{bmatrix} 1 & 2 & 2 & 1 \\ -1 & -2 & -2 & -1, \end{bmatrix} / 12 \tag{23}$$

in which each matrix entry gives the value of a pixel of the PSF, and the arrangement of the matrix entries corresponds to the spatial arrangement of the corresponding pixels in the PSF. The other filters were obtained by rotating this base filter, with bicubic interpolation, by angles multiple of 45°. In the figures that we show ahead, the estimated image was first subjected to the affine transformation mentioned in Section III, and was then saturated to the maximum and minimum values of the blurred image.

The blurred images were normalized so that black corresponded to $-0.5$ and white (or maximum intensity, in the case of color channels of a color image) corresponded to 0.5. Parameter $\epsilon$ was set to 0.002.

The sequence of values of $\lambda$ was a geometric progression ($\lambda_{n+1} = \lambda_n / r$), initialized at $\lambda_1 = 2$. The values that were used for $r$ are given ahead for each specific case. For real-life photos, the iteration on $\lambda$ was stopped on the basis of visual evaluation. In the synthetic experiments, we used the ISNR for deciding when to stop the iteration. The selection of the stopping point was rather important for noisy blurred images because, after the "optimal" point, the estimated image quickly became degraded with noise. For non-noisy blurred images, the method typically stopped progressing after a certain value of $\lambda$. After that value, the choice of the stopping point had almost no influence on the result.

All experiments were performed using the same sequence of values for parameter $q : 0.8, 0.8, 0.6, 0.6, 0.6, 0.6, 0.4, \ldots, 0.4$. The sequences of values of $\lambda$ and $q$ were experimentally found to be adequate for a wide variety of images and blurs, and can be used, without change, in most situations.

The support of the estimate of the blurring filter was limited to a square of size $s \times s$ pixels, chosen to be slightly larger than the size of the actual blur (the value of $s$ is given ahead for each case). We used a safety zone (see Section II-C) with a width of three pixels around the support of the filter.

In most cases, the cost function was quadratic in $h$, and the optimization relative to $h$ was performed by a relatively fast method (conjugate gradients, with 100 iterations for each value of $\lambda$). In the cases in which this function was not quadratic (the cases in which a TV regularizer was used on the blurring filter and the cases in which we used a parametric model for the filter), gradient descent with adaptive step sizes [54] was used, because it can easily deal with strongly nonquadratic functions and still is relatively fast. The optimization relative to the deblurred image $x$ was also performed by gradient descent with adaptive step sizes (150 iterations for each value of $\lambda$), because the cost function is a strongly nonquadratic, nonsmooth function of $x$, and this method can also easily deal with nonsmooth functions. The numbers of iterations mentioned above were experimentally chosen so as to achieve a good convergence of the corresponding optimizations. These numbers are not crucial: they could have been increased without negatively impacting the deblurring results, but such an increase would obviously also have increased the running time of the algorithm, with no significant advantage.

On an Intel Core 2 Duo system running at 2 GHz, programmed in Matlab and running on only one of the chip's processors, an iteration of the method, corresponding to one value of $\lambda$, took, for monochrome images of size $256 \times 256$, about 30 seconds, when conjugate gradients were used for the optimization relative to $h$. For color images, each iteration took about 70 seconds, also with conjugate gradient optimization of $h$.

The total deblurring time depended on the number of iterations in $\lambda$, which depended on the ratio $r$ and on the stopping point of the $\lambda$ sequence. Experiments with blurred monochrome photos (Section IV-B) were processed using $r = 3$ and with around ten values of $\lambda$, having taken about 5 min to be processed. Synthetic experiments were processed using a lower ratio of $r = 1.5$ and approximately 55 and 23 iterations for non-noisy and noisy experiments, respectively, having taken about 28 and 12 min, respectively. Results with only slightly lower quality were obtained with a ratio of $r = 3$ and with a somewhat higher stopping value for $\lambda$ (see [11]), resulting in a much shorter processing time of 7.5 min for non-noisy images. For color images, all these times were multiplied by about 2.5.

### A. Synthetic Degradations

In this section we first describe the main experiment, which was intended at showing that the proposed method can effectively deal, in a blind way, with a large variety of images and of blurs. After that, we describe additional tests that were performed to check some other aspects of the method's performance. All of these experiments were run with a ratio of $r = 1.5$ in the sequence of values of $\lambda$. The iteration was run up to $\lambda_{55} = 6.2 \times 10^{-10}$, and the best stopping iteration was chosen based on the values of the ISNR measure, computed as described in Section III.

The main experiment was performed with the five grayscale images shown in Fig. 6. Each image was blurred with each of the blurring filters shown in Fig. 7 (for a better visualization, these filters, as well as the filter estimates to be presented further ahead, are shown with a band of zero-valued pixels around them). All filters were normalized to a DC gain of 1. The PSF of filter #1 is a uniform-intensity circle, and simulates an out-of-focus blur. Filter #2 simulates a linear, uniform motion blur. The PSF of filter #3 is a uniform-intensity square. Filter #4 was formed by choosing random pixel intensities with a uniform distribution in $[-0.3, 0.7]$, postnormalized to a DC gain of 1. Filter #5 simulates an irregular motion blur. Filter #6 corresponds to a circular motion blur, and was chosen because its frequency response has somewhat a nonlowpass character and, therefore, is rather different from the most common blurs. Filter #7 is Gaussian, with a standard deviation of two pixels.

Filter #1 had a radius of $l$ pixels, and filters #2, #3, and #4 had a size of $l \times l$ pixels. We used different values of $l$ for different cases: $l = 9$ for "Lena", "Barbara" and "Testpat1", and $l = 11$ for "Cameraman" and "Satellite". For the "Lena," "Barbara," and "Testpat1" images, the size of the estimated filters was set to $s \times s$ pixels, with $s = 15$; for "Cameraman" and "Satellite" we used $s = 17$, due to the larger size of the blurs. No constraints were imposed on the estimated filters. Each blurred image was used both without noise and with Gaussian i.i.d. noise at a *blurred signal to noise ratio* (BSNR) of 30 dB.

Figs. 8(b) to 10(b) show a sample of the 70 blurred images used in this experiment, and the second rows of Figs. 8–10 show

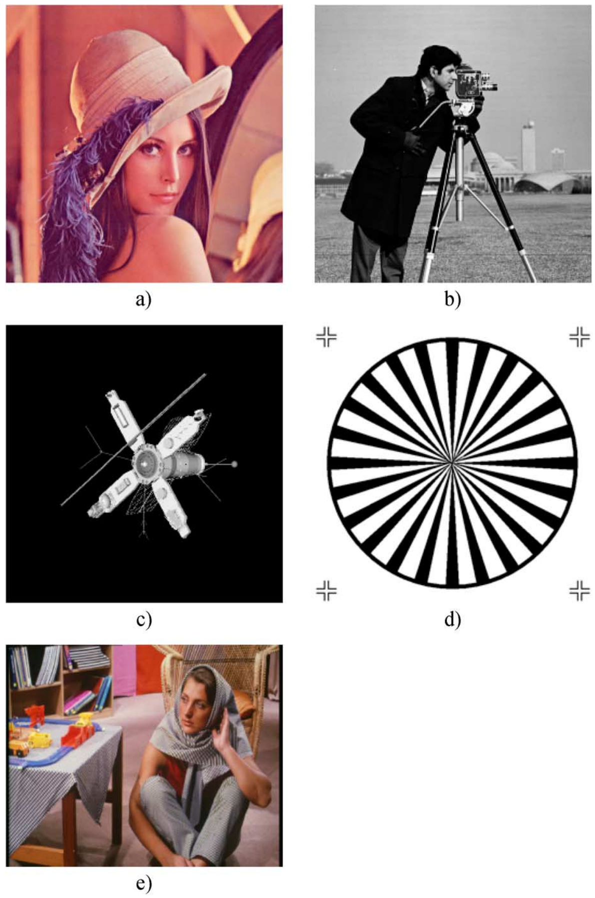


Fig. 6. Set of images used for synthetic experiments. (a) “Lena”. (b) “Cameraman”. (c) “Satellite”. (d) “Testpat1”. (e) “Barbara”.

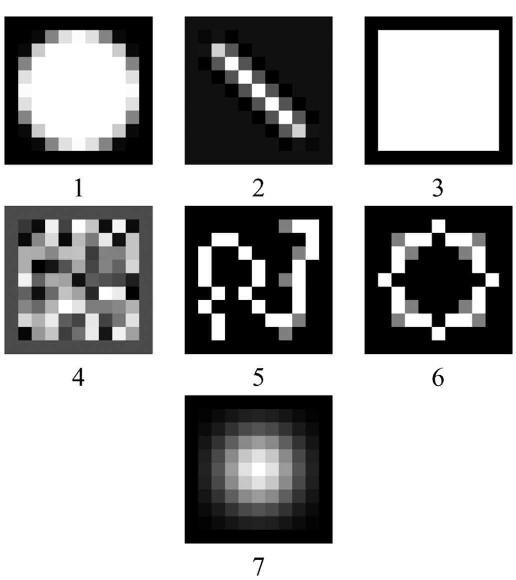


Fig. 7. Set of blurring filters used in synthetic experiments. #1: Out-of-focus blur. #2: Linear motion blur. #3: Uniform square. #4: Random square. #5: Nonlinear motion blur. #6: Circular motion blur. #7: Gaussian.

deblurred images that were obtained by the method. Table II gives a summary of the results, in terms of ISNR. Detailed results are given in Appendix A. We can see that the method yielded, in almost all cases, a significant improvement in image

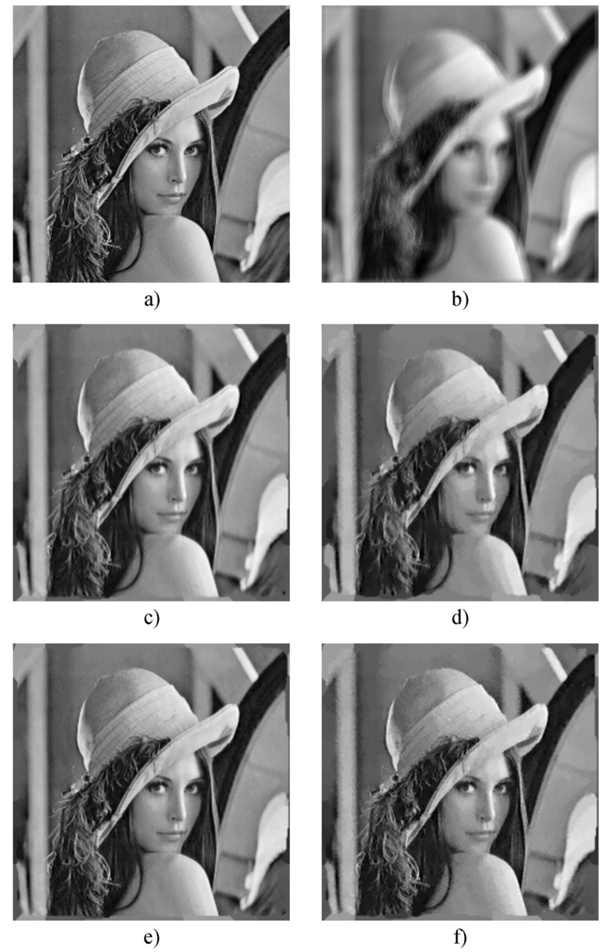


Fig. 8. “Lena” image with blur #5. (a) Sharp image. (b) Blurred image. Next rows: Deblurred images. Left: Without noise. Right: With noise. Second row: Without filter constraints or regularization. Third row: Using TV regularization on the blurring filter.

quality. The blurring filters also were reasonably recovered, especially when there was no noise (see Figs. 11 and 12). As mentioned in Section II-B, the images recovered in noisy situations had a slight piecewise-constant character.

The worst results corresponded to the Gaussian blur (#7). This has a simple explanation. Although, visually, the Gaussian filter does not look worse than, say, the square or the circular ones, its frequency response decays much faster than those of the other filters. At the spatial frequency of $f_{\max} = 1/(2 \text{ pixels})$, which is the maximum frequency allowed by the sampling theorem, the Gaussian filter presents an attenuation above 150 dB. At the frequency $f_{\max}/2$, the attenuation is of more than 40 dB. This means that the filter eliminates essentially all the high frequency content from the original image, and there is no way to recover it (recall that, even without added noise, the blurred images do have noise due to rounding errors).

Our second test concerned the use of constraints on the blurring filter. In this test we used all blurring filters except #4 and #5, for which no simple constraints existed. For filter #1, we used as parametric model a uniform circle, with the diameter as parameter. Since the diameter could take any real, positive

Fig. 9. "Barbara" image with blur #2. (a) Sharp image. (b) Blurred image. Next rows: Deblurred images. Left: Without noise. Right: With noise. Second row: Without filter constraints or regularization. Third row: With constraints.

value, the filter's PSF was obtained by computing the pixel intensities according to the fraction of each pixel that was covered by the circle with the prescribed diameter; the gradient of the cost function relative to the diameter was computed taking this model into account. For filter #2, the constraint was symmetry relative to the central pixel. For filters #3, #6, and #7, the constraint was symmetry relative to the four axes that make angles multiple of 45° with the horizontal axis. The tests were run on all images of Fig. 6, without noise and with noise at 30-dB BSNR. Figs. 8(d) and (f) to 10(d) and (f) show a sample of the noisy estimates (we do not show the noisy blurred images because, visually, they are almost indistinguishable from the non-noisy ones). Table III shows a summary of the ISNR values (the complete list of values is given in Appendix A). It can be seen that, in most cases, the use of constraints improved the quality of the results. In some cases the improvement was quite impressive, and, on the other hand, in a very few cases, there was a very slight decrease in quality.

As mentioned above, we used a parametric model as constraint for filter #1. Therefore, in this case, the estimation of the blurring filter actually consisted of the estimation of its single parameter (the diameter). In the noiseless case, the estimated diameter values were of 8.84, 10.96, 10.96, 8.56, and 8.94 pixels, respectively, for the five images. These values compare well with the true diameter values of 9, 11, 11, 9, and 9 pixels, respectively.

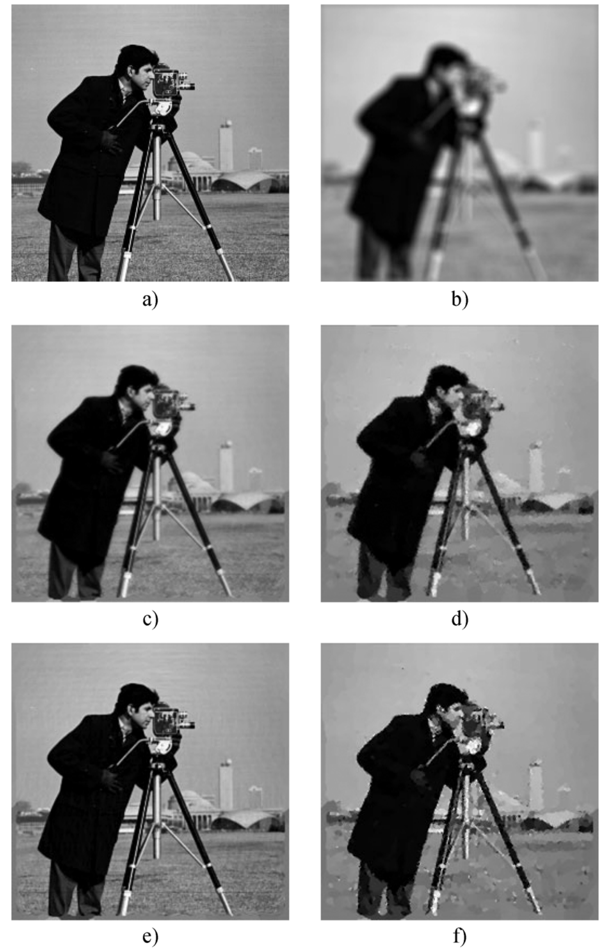

Fig. 10. "Cameraman" image with blur #1. (a) Sharp image. (b) Blurred image. Next rows: Deblurred images. Left: Without noise. Right: With noise. Second row: Without filter constraints or regularization. Third row: With constraints.

TABLE II
SUMMARY OF THE ISNR VALUES OBTAINED WITH OUR METHOD, WITH NO CONSTRAINTS AND NO REGULARIZATION ON THE ESTIMATED FILTER. EACH ENTRY GIVES THE AVERAGE OF THE ISNRS OBTAINED FOR THE FIVE TEST IMAGES, UNDER THE INDICATED CONDITIONS

| Blur | #1 | #2 | #3 | #4 | #5 | #6 | #7 |
|---|---|---|---|---|---|---|---|
| Without noise | 6.48 | 5.38 | 6.43 | 5.87 | 6.29 | 5.82 | 3.21 |
| With noise | 3.69 | 4.19 | 3.61 | 5.46 | 5.50 | 6.34 | 1.99 |

A third test, performed only on the "Lena" image, concerned the use of a TV regularizer on the estimate of the blurring filter's PSF (without constraints). For this test, we used the cost function of (18), with a ratio $\lambda_h/\lambda = 100$. The ISNR values, given in Table VI in Appendix A, show that the use of the regularizer improved the SNR, in all cases, relative to the estimation without constraints. Although this regularizer yielded an improvement in the results, it was used here only as an example. Other regularizers may be more useful for specific situations.

The ISNR values attained in the tests described so far can be considered rather good, taking into account that the method under test is blind. In fact, these ISNR values are relatively close to the values attained by state of the art nonblind methods. For

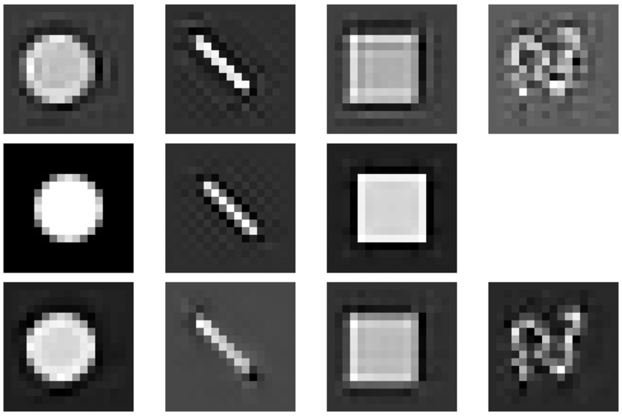

Fig. 11. Filter estimates, for non-noisy experiments. First row: Without restrictions. Second row: With restrictions. Third row: With TV regularization.

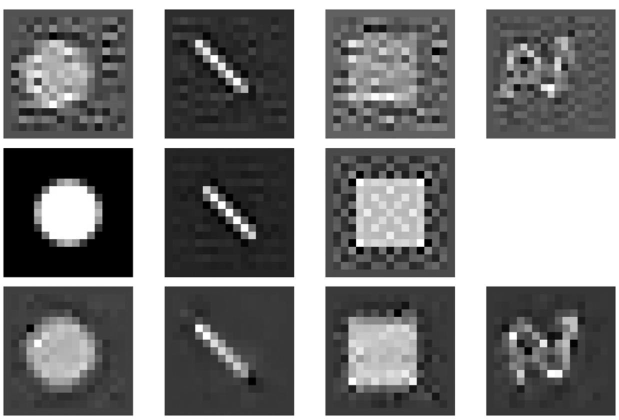

Fig. 12. Filter estimates, for noisy experiments. First row: Without restrictions. Second row: With restrictions. Third row: With TV regularization.

TABLE III
COMPARISON OF THE RESULTS OBTAINED WITHOUT AND WITH CONSTRAINTS ON THE ESTIMATED FILTER. EACH ENTRY GIVES THE AVERAGE OF THE ISNRS OBTAINED FOR THE FIVE TESTED IMAGES, UNDER THE INDICATED CONDITIONS. THE BEST RESULT FOR EACH CASE IS SHOWN IN BOLD

| | Blur | #1 | #2 | #3 | #6 | #7 |
|---|---|---|---|---|---|---|
| **Without noise** | **Without Constraints** | 6.48 | 5.38 | 6.43 | 5.82 | 3.21 |
| | **With Constraints** | **12.01** | **8.61** | **10.22** | **11.94** | **4.72** |
| **With noise** | **Without Constraints** | 3.69 | 4.19 | 3.61 | 6.34 | 1.99 |
| | **With Constraints** | **5.34** | **6.64** | **5.25** | **10.09** | **3.14** |

example, the recent work [23] presents the ISNR values attained by several state of the art nonblind methods on the restoration of the "Lena" image blurred with a $9 \times 9$ uniform blur, with 30 dB BSNR. Those values range from 5.58 dB, corresponding to the method from [55], to 6.1 dB, attained by the method from [23]. Our method blindly attained, in the same problem, about 1.5 dB less (4.07 dB without constraints, 4.47 dB with constraints, and 4.27 dB with TV regularization). The majority of the nonblind deblurring approaches have been tested with blurs produced with circular convolutions, while our method used linear convolutions. The circular convolutions introduce, in the original image, an artificial, long, straight, horizontal edge, corresponding to making the image top and bottom adjacent to each other, and also introduce a similar artificial vertical edge. The presence of these artificial edges would have helped our method to better estimate the blurring filter, slightly improving the deblurring results (evidence of this can be found in some results presented later, in Table V, in which we find improvements between 0.03 dB and 1.18 dB, for our method, due to the use of circular convolutions).

a) b) c) d)

Fig. 13. Deblurring of a color image. (a) Sharp image. (b) Image degraded with blur #2 and 30 dB of noise. (c) Image estimate without noise. (d) Image estimate with noise.

We performed another test to check the method's performance on color images. We used the cost function given by (13). For this test, we used the color "Lena" and "Barbara" images, with all the blurs described above, without constraints. Figs. 13 and 14 show some results. The ISNR of the results is shown in Tables VI and IX, in Appendix A. The results on color images were, in general, slightly better (and, in a few cases, significantly better) than those for grayscale images. This is not surprising, since the three color channels of color images contain more information about the blur than the single channel of grayscale images.

For testing the deblurring of multiframe images, as described in Section II-E, we used the "Lena" image with two blurred frames, both with motion blurs. Both blurs had a length of 11 pixels, and they had directions that made angles of 45° and 135°, respectively, with the horizontal axis. We used the noiseless blurred images, and also noisy images with BSNRs of 40 and 30 dB. To assess the advantage of using multiframe deblurring, we compared the images recovered in multiframe mode with the images recovered, in single-frame mode, from each of the two frames. Fig. 15 shows the results for the 30 dB noise level. Table IV gives the ISNR values. We can see that the multiframe method was advantageous in situations with noise, but had no clear advantage in the situation without noise.

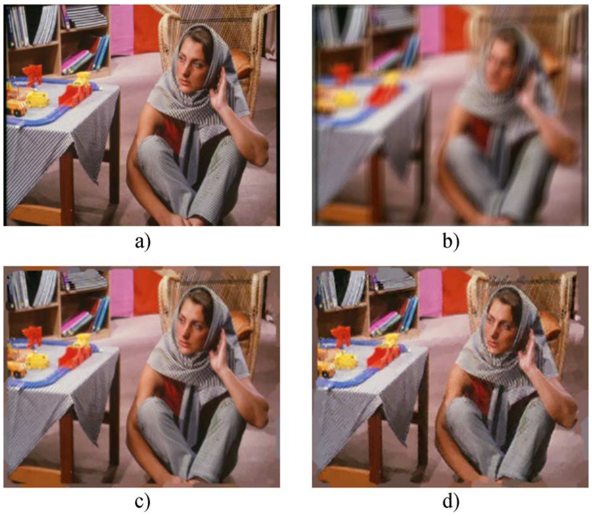


Fig. 14. Deblurring of a color image. (a) Sharp image. (b) Image degraded with blur #5 and 30 dB of noise. (c) Image estimate without noise. (d) Image estimate with noise.

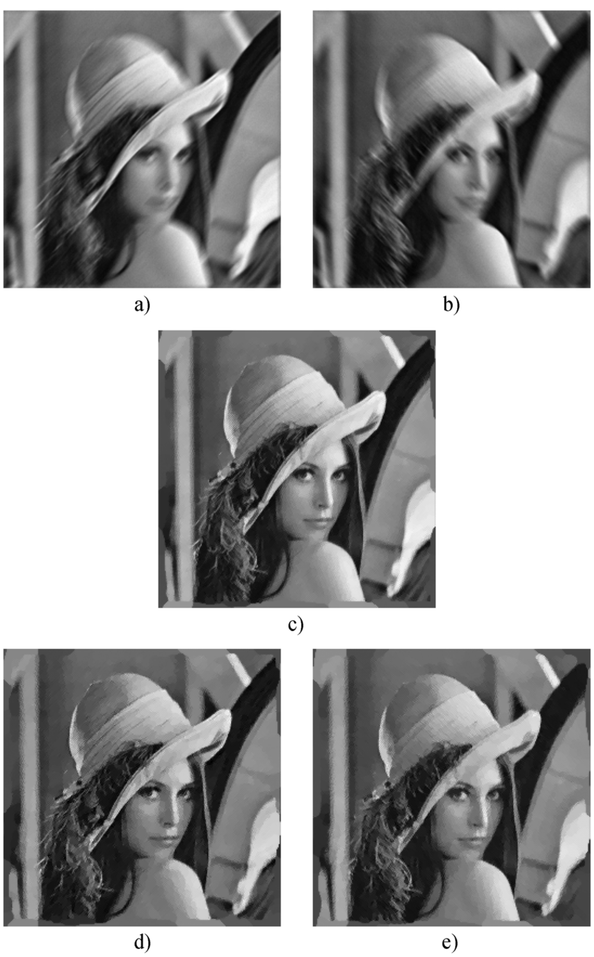


Fig. 15. Multiframe and single-frame estimates obtained with 30-dB BSNR. First row: Degraded images. Second row: Multiframe image estimate. Third row: Single-frame image estimates.

## B. Blurred Photos

Besides testing the method on synthetic degradations, we also applied it to real-life blurred photos. We used two different color

TABLE IV

MULTIFRAME PERFORMANCE VERSUS SINGLE-FRAME PERFORMANCE ("LENA" IMAGE WITH MOTION BLURS OF 11 PIXELS). THE LAST COLUMN SHOWS THE SNR (IN dB) OF THE DEBLURRED IMAGES, RELATIVE TO THE ORIGINAL SHARP ONE. THE BEST RESULTS FOR EACH CASE ARE SHOWN IN BOLD

| Mode | Blur directions | Noise BSNR | SNR of the result |
|---|---|---|---|
| Single-frame | 45° | | 15.50 |
| Single-frame | 135° | - | **16.22** |
| Multi-frame | 45°,135° | | 16.09 |
| Single-frame | 45° | | 15.35 |
| Single-frame | 135° | 40dB | 15.14 |
| Multi-frame | 45°,135° | | **15.88** |
| Single-frame | 45° | | 14.12 |
| Single-frame | 135° | 30dB | 13.83 |
| Multi-frame | 45°,135° | | **15.19** |

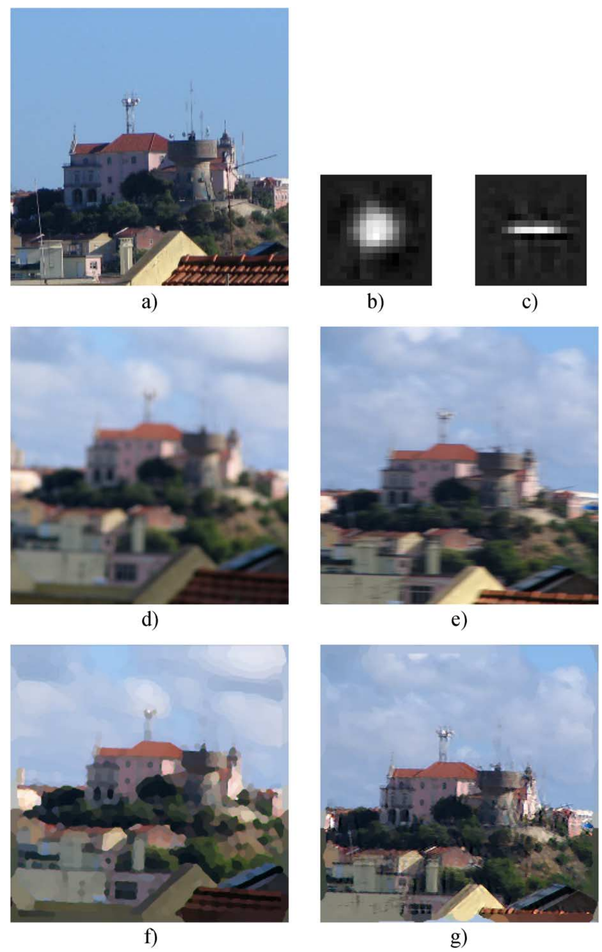


Fig. 16. Results with real-life blurred photos. (a) Sharp photo of the scene. (b), (c) Filter estimates. Next rows: Left-out-of-focus blur. Right-motion blur. Second row: Blurred photos. Third row: Deblurred images.

scenes [Figs. 16(a) and 17(a)]. The corresponding grayscale images were also processed (grayscale results are presented in Section IV-C).

We addressed two kinds of real-life degradations: the pictures in Fig. 16(d) and Fig. 17(c) were purposely taken with the camera wrongly focused, while in Fig. 16(e) the camera was purposely rotated in the horizontal direction while the photo was being taken, to produce a motion blur. The photos of Fig. 16 were taken with a Canon S1 IS camera, and were

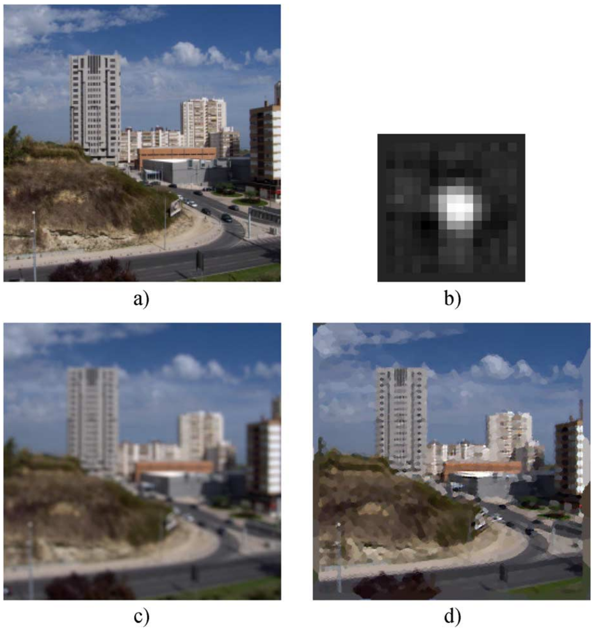


Fig. 17. Results with an actual blurred photo. (a) Sharp photo of the scene. (b) Filter estimate. (c) Blurred photo. (d) Deblurred image.

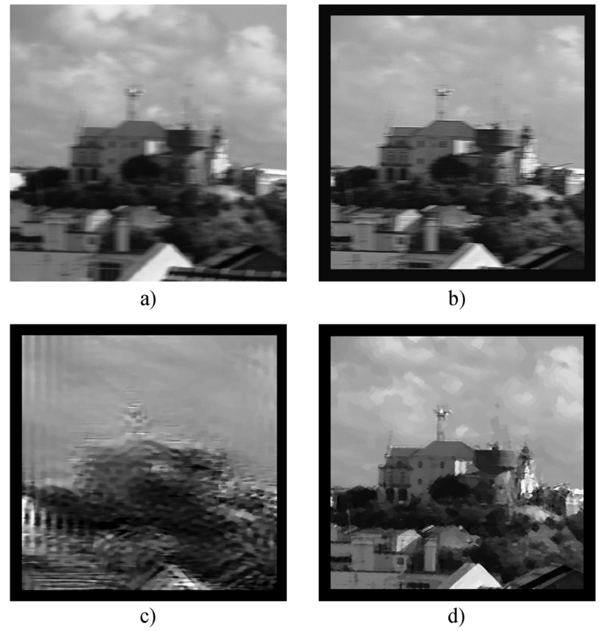


Fig. 18. Deblurring of an actual photo with several methods. (a) Blurred photo. (b) Image deblurred with APEX. (c) Image deblurred with the YK method. (d) Image deblurred with our method.

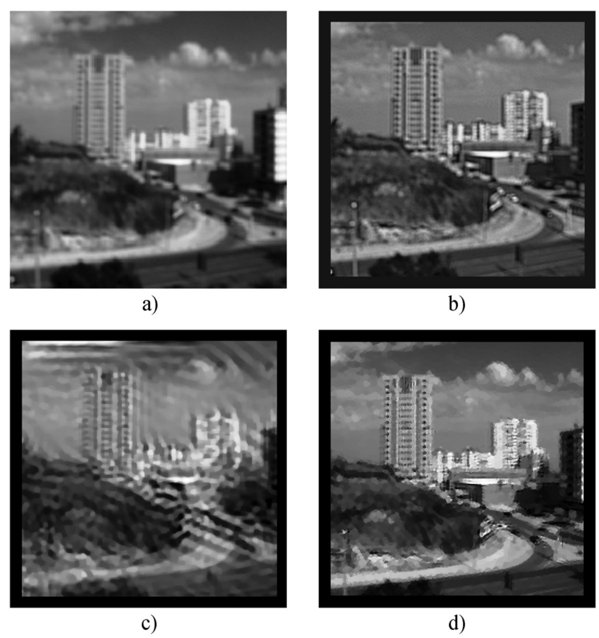


Fig. 19. Deblurring of an actual photo with several methods. (a) Blurred photo. (b) Image deblurred with APEX. (c) Image deblurred with the YK method. (d) Image deblurred with our method.

coded in JPEG format (this camera cannot save images in RAW format). The photos of Fig. 17 were taken with a Panasonic DMC-FZ18 camera, and were coded in RAW format (i.e., using actual sensor data, after the demosaicing that interpolated colors among pixels).

The noise that was present in the photos was quite significant. The cameras that we used have small image sensors (about 1 cm diagonal), which yield images with a significant amount of noise (much larger than the 30 dB that we used in the synthetic experiments). In the images with full sensor resolution, the noise was too high to allow any significant improvement by means of the deblurring method. In order to simulate cameras with larger sensors, we reduced the resolution of the camera images by averaging in squares of $6 \times 6$ pixels, for the Canon camera, and of $9 \times 9$ pixels, for the Panasonic camera. This reduced the noise to acceptable levels.

The size of the blur estimate was limited to a square of size $15 \times 15$ pixels. We used a sequence of $\lambda$ values with a ratio of $r = 3$, and truncated this sequence at $\lambda_9 \approx 3.05 \times 10^{-4}$ for the experiments of Fig. 16 and at $\lambda_{10} \approx 1.02 \times 10^{-4}$ for the experiment of Fig. 17.

All the recovered images were significantly sharper and had more visible details than the blurred ones, even though they had somewhat a "patchy" look, corresponding to somewhat a piecewise-constant character. As had happened with the synthetic degradations, the restoration was slightly better for color photos than for monochrome ones [compare Fig. 16(f) with Fig. 18(d) and Fig. 17(d) with Fig. 19(d)].

The results obtained with these photos were of lower visual quality than those obtained with the synthetic blurs. Two of the reasons for this probably were as follows.

- The blurs that were present in the photos probably did not exactly follow the model of (1). One of the main reasons may have been the presence of nonlinearities in the image acquisition. It is known that image sensors may not be perfectly linear, due to the presence of anti-blooming circuits, for example. Furthermore, in the case of the Canon camera, for which we did not have access to RAW data, we suspect that the camera also performed some nonlinear operations like denoising, sharpening and gamma compensation. In an actual application (for example, if deblurring is to be incorporated in the image processing performed in the camera itself), it should be possible to avoid, or to compensate for, these nonlinearities.

TABLE V
ISNR VALUES (IN DECIBELS) OBTAINED FOR SEVERAL METHODS. FOR EACH DEGRADATION, THE BEST RESULTS ARE SHOWN IN BOLD

| Lena 256 × 256 | | **Linear convolutions** | | | | | **Circular convolutions** | | | **DCT-based convolutions** |
|---|---|---|---|---|---|---|---|---|---|---|
| | | **Our method** | | | **APEX** | **YK** | **Our method** | | **APEX** | **YK** |
| **BSNR** | **Blur** | **Without constraints** | **With constraints** | **With TV prior** | | | **Without constraints** | **With constraints** | | |
| - | # 3 | 7.81 | **9.55** | 8.78 | 0.39 | -6.89 | 8.74 | **9.58** | 0.52 | 0.43 |
| | # 4 | 7.26 | - | **9.82** | 0.39 | -1.39 | **6.29** | - | 0.56 | -4.14 |
| | # 5 | 8.87 | - | **11.21** | 0.38 | -3.59 | **10.05** | - | 0.55 | -0.27 |
| | # 7 | 3.93 | **3.99** | **3.99** | 1.69 | -7.99 | 4.03 | 4.12 | **4.21** | -3.02 |
| 30dB | # 3 | 4.07 | **4.47** | 4.27 | 0.60 | - | 4.26 | **4.48** | 0.74 | - |
| | # 4 | 5.99 | - | **6.74** | 0.59 | - | **6.03** | - | 0.59 | - |
| | # 5 | 6.76 | - | **8.04** | 0.53 | - | **7.56** | - | 0.69 | - |
| | # 7 | 2.23 | **2.33** | 2.29 | 1.77 | - | 2.40 | **2.50** | 2.29 | - |

- The noise produced by image sensors is not Gaussian and (probably more important) its intensity is not independent from the image's intensity, contrary to the assumptions of our method.

### C. Comparison With Other Methods

We compared our method with two other methods from the literature: APEX [33], [34] and the method from [2] (which we shall call YK method). These were the only two methods for which we were able to obtain implementations. APEX was simple enough for us to implement ourselves within useful time, and the authors of the YK method kindly provided a demo, coded in C++.

The APEX method [33], [34] is quite fast, but is limited to blurs which belong to the Levy family. This is a family with just two parameters, in which all functions have circular symmetry, and which encompasses the Gaussians. The method has two regularizing parameters ($M$ and $s$), whose values we have set to those recommended by the author ($M = 500$ and $s = 0.01$). The method has two further parameters (designated by $A$ and $t$, respectively). For $A$, we used the values 2.00, 2.25, 2.50, ..., 7.75, 8.00, which cover the recommended interval. Parameter $t$ can be varied between 1 and 0, $t = 1$ corresponding to the blurred image, and $t = 0$ to a "completely deblurred" one. We used the values 1, 0.75, 0.5, 0.25, and 0. For synthetic blurs, the ISNR values were computed for all combinations of values of $A$ and $t$, and the best combination was selected. For real-life blurred photos, the best pair was chosen by visual inspection, since no ISNR values could be computed.

The YK method does not constrain the blur PSF, but assumes that it is piecewise smooth (and, from the comments made in [2], one can see that the method has some bias toward piecewise constant PSFs). The method has four parameters that must be manually chosen. We started by trying the values used in [2] but, with our blurred images, this produced results with very strong oscillatory artifacts. After several tests, we chose the following values, which seemed to produce the results with fewest artifacts: 1 and 2000 for the regularizing parameters of the image and of the PSF, respectively; 0.1 and 0.001 for the threshold parameters of the diffusion coefficients of the image and of the PSF, respectively. We should note that our tests were severely limited by the fact that the deblurring of each image, with this method, took about 12 h, despite the fact that the method was coded in C++. Besides preventing us from doing a more extensive search of parameter values, this also prevented us from testing the method on noisy synthetic degradations.

The APEX method uses circular convolutions in the blurring model. The YK method uses convolutions computed from products of discrete cosine transforms (DCTs). Our method normally uses linear convolutions, but can easily be modified to use circular or DCT-based convolutions. We tested all methods with degradations produced with linear convolutions, and also tested both our method and APEX with degradations produced with circular convolutions. Given the very poor results (to be seen ahead) obtained by the YK method with DCT-based convolutions, and the limited time that was available to us, we did not consider it necessary to test our method with DCT-based convolutions.

For the comparison, we used both synthetic and real-life degradations. The synthetic degraded images were obtained from the grayscale "Lena" image, with blurs #3, #4, #5, and #7. Blur #7 is Gaussian and, therefore, is within the family of blurs for which the APEX method is appropriate. Blur #3 is piecewise constant and, therefore, appears to be appropriate for the YK method. Blur #5 may also be considered piecewise constant. Blur #7 is smooth and, therefore, appears to be at least partially appropriate for that method, too. The real-life degraded images that we used were grayscale versions of two of the photos presented above, one with a motion blur and the other with an out-of-focus blur. The size of the estimated blurring filter was limited to 15 × 15, both in our method and in the YK one. APEX does not assume a limited size of the blurring filter.

Table V shows the ISNR values obtained with the synthetic degradations. We can see that, with linear convolutions, our method clearly surpassed the other two methods. APEX only yielded a significant improvement in the image quality for the Gaussian blur, as expected. When we used blurs produced with circular convolutions, which are the most appropriate ones for APEX, the results of our method improved slightly, in all cases but one. The results of APEX improved significantly in the case of the Gaussian blur (especially without noise), and improved only slightly for the other blurs. The APEX method only surpassed ours in the case of the Gaussian blur without noise. Even in that case, the advantage of APEX over our method was sonly slight, despite the fact that the Gaussian blur is within the class of blurs for which APEX was designed, and that APEX was only estimating two parameters, while our method was estimating 225.

The performance of the YK method was rather poor, which was somewhat a surprise to us. A possible explanation for the difference between these results and the ones presented in [2] is that, while the tests described in that reference involved the estimation of PSFs with up to 49 parameters, the tests performed by us involved PSFs with $15 \times 15 = 225$ pixels. However, we should note that, with four parameters to be manually chosen in that method, it is hard to find a good set of values, and it may be questioned whether the method really should be considered blind. Still another explanation could be the fact that, as the algorithm's authors themselves say, the algorithm suffers quite severely from the problem of local minima.

Figs. 18 and 19 show the results obtained with actual photos. The APEX method produced almost no improvement in the motion blur case, and produced a moderate improvement in the out-of-focus blur case. Both of these results are understandable: Motion blurs are well outside the family of blurs for which the method is appropriate, and the PSF of the out-of-focus blur that exists in the second photo seems not to be too far from a Gaussian [see Fig. 17(b)], and probably is close to the family of blurs for which APEX is appropriate. Nevertheless, the result produced by our method was sharper, even in this case.

The results produced by the YK method show the kind of problem that affected many of the results obtained with that method: there were strong oscillatory artifacts, even with the parameter values that we used. We should note, however, that, for the test images sent by the method's authors, the method did yield results similar to the ones published by them. This gives us some confidence that the method was correctly applied. We have already speculated above about possible reasons for the poor results obtained in our tests with that method.

### D. Final Comments

We stress that, although our method involves a few parameters, only one of them is crucial (and only for noisy blurs): the stopping point of the iteration. In fact, our tests have shown that the same sequences of values of $\lambda$ and of $q$ yielded good results for a wide range of images and of blurs.[1] Therefore, these sequences can be fixed *a priori*, without knowing which image and which blur are to be handled. This being said, we should note that, by tuning these parameters, somewhat better results can be obtained than the ones that we have shown in this paper. In several practical applications, it may be quite possible to pretune such parameters. For example, in a digital camera, pretuned values can exist for different apertures, focal lengths, etc.

The choice of the stopping point of the iteration is not crucial for non-noisy images, as we said above. For noisy images, we do not have any good stopping criterion yet. The choice of the stopping point is very similar, in character, to the choice of the $t$ value in the APEX method, and is a known difficult problem, even for nonblind methods, for which several solutions have been proposed (e.g., [56]). None of these solutions showed to be robust enough for our application.

[1]We have used two different values of the ratio $r$ in different situations, but the smaller value can be used in all cases, with the only disadvantage of needing more iterations and, therefore, taking a longer time, to reach the final value.

## V. Conclusion

We have presented a method for blind image deblurring. The method differs from most other existing methods by only imposing weak restrictions on the blurring filter, being able to recover images which have suffered a wide range of degradations. Good estimates of both the image and the blurring operator are reached by initially considering the main image edges, and progressively handling smaller and/or fainter ones. The method uses an image prior that favors images with sparse edges, and which incorporates an edge detector that was specially developed for this application. The method can handle both unconstrained blurs and constrained or parametric ones, and it can deal with both single-frame and multiframe scenarios.

Experimental tests showed good results on a variety of images, both grayscale and color, with a variety of synthetic blurs, without and with noise, with real-life blurs, and both in single and in multiframe situations. The use of information on the blurring filter and/or of multiframe data, when available, typically led to improvements in the quality of the results.

We have adapted the ISNR measure to the evaluation of the restoration performance of BID methods. The restoration quality of our method was visually and quantitatively better than those of the other methods with which it was compared.

So far, whenever the blurred image has noise, the processing has to be manually stopped, by choosing the iteration which yields the best compromise between image detail and noise or artifacts. An automatic stopping criterion will obviously be useful. This is a direction in which further research will be done.

The method can be extended in other directions: For example, 1) to address problems in which we aim at super-resolution, possibly combined with deblurring, and 2) to deblur images containing space-variant blurs (for example, a sharp scene containing one or more motion-blurred objects, or a scene containing objects at different distances from the camera, with different out-of-focus blurs). This latter extension has already shown useful results [8].

Finally, on a more theoretical level, but with possible practical implications, is the problem that we mentioned above, that the best deblurring solutions generally do not correspond to the global minimum of the cost function. This apparently means that a more appropriate cost function should exist. If it were found, it would probably lead to a better deblurring technique, both in terms of speed and of the quality of the results. This clearly is an important research direction.

## Appendix

### A. Tables

Tables VI–X show detailed ISNR values from several of the tests mentioned in this paper.

TABLE VI
ISNR (in Decibels) Computed for Experiments Performed With the "Lena" Image. For Each Degradation of the Grayscale Image, the Best Value is Shown in Bold

| Lena 256 × 256 | | Grayscale | | | Color |
|---|---|---|---|---|---|
| **BSNR** | **Blur** | **Without const.** | **With const.** | **TV prior** | **Without constraints** |
| - | # 1 | 7.87 | **9.68** | 9.66 | 10.75 |
| | # 2 | 7.09 | 6.92 | **7.83** | 8.83 |
| | # 3 | 7.81 | **9.55** | 8.78 | 9.70 |
| | # 4 | 7.26 | - | **9.82** | 12.37 |
| | # 5 | 8.87 | - | **11.21** | 10.00 |
| | # 6 | 8.50 | **11.51** | 9.59 | 9.38 |
| | # 7 | 3.93 | **3.99** | **3.99** | 4.62 |
| 30 dB | # 1 | 4.42 | 4.15 | **4.45** | 5.05 |
| | # 2 | 5.56 | 5.33 | **5.74** | 6.15 |
| | # 3 | 4.07 | **4.47** | 4.27 | 4.91 |
| | # 4 | 5.99 | - | **6.74** | 6.85 |
| | # 5 | 6.76 | - | **8.04** | 8.02 |
| | # 6 | 7.35 | **8.93** | 7.36 | 8.24 |
| | # 7 | 2.23 | **2.33** | 2.29 | 3.15 |

TABLE VII
ISNR (in Decibels) Computed for Experiments Performed With the "Cameraman" Image. For Each Degradation, the Best Value is Shown in Bold

| Cameraman 256 × 256 | No noise | | BSNR at 30dB | |
|---|---|---|---|---|
| **Blur** | **Without constraints** | **With constraints** | **Without constraints** | **With constraints** |
| # 1 | 6.32 | **14.47** | 4.27 | **5.20** |
| # 2 | 4.87 | **6.60** | 4.15 | **5.24** |
| # 3 | 5.51 | **6.72** | 4.07 | **4.24** |
| # 4 | **5.62** | - | **4.90** | – |
| # 5 | **6.18** | - | **6.20** | – |
| # 6 | 8.06 | **8.30** | 6.64 | **7.42** |
| # 7 | 2.72 | **3.24** | 1.81 | **2.30** |

TABLE VIII
ISNR (in Decibels) Computed for Experiments Performed With the "Satellite" Image. For Each Degradation, the Best Value is Shown in Bold

| Satellite 256 × 256 | No noise | | BSNR at 30dB | |
|---|---|---|---|---|
| **Blur** | **Without constraints** | **With constraints** | **Without constraints** | **With constraints** |
| # 1 | 8.60 | **13.60** | 5.42 | **6.34** |
| # 2 | 7.07 | **8.37** | 5.99 | **7.38** |
| # 3 | 8.76 | **10.24** | 5.14 | **5.33** |
| # 4 | **7.76** | - | **8.17** | – |
| # 5 | **8.72** | - | **7.08** | – |
| # 6 | 8.56 | **12.23** | 7.19 | **10.92** |
| # 7 | 3.29 | **3.44** | 2.33 | **2.75** |

### *B. Gradients*

Gradients of the method's cost function are explicitly given in this appendix. The cost function is

$$C(x,h) = \frac{1}{2}\|y - h * x\|_2^2 + \lambda \sum_i \left[f_i(x) + \epsilon\right]^q \qquad (24)$$

TABLE IX
ISNR (in Decibels) Computed for Experiments Performed With the "Barbara" Image. For Each Degradation of the Grayscale Image, the Best Value is Shown in Bold

| Barbara 230 × 288 | | Grayscale | | Color |
|---|---|---|---|---|
| **BSNR** | **Blur** | **Without constraints** | **With constraints** | **Without constraints** |
| - | # 1 | **5.66** | 4.31 | 5.83 |
| | # 2 | 4.77 | **5.68** | 5.12 |
| | # 3 | 6.22 | **7.43** | 6.19 |
| | # 4 | **4.56** | - | 7.04 |
| | # 5 | **6.17** | - | 6.87 |
| | # 6 | 6.07 | **13.43** | 6.06 |
| | # 7 | 2.99 | **3.01** | 3.02 |
| 30 dB | # 1 | **3.26** | 2.67 | 3.99 |
| | # 2 | 3.37 | **4.19** | 4.26 |
| | # 3 | **3.73** | 3.72 | 4.18 |
| | # 4 | **4.56** | - | 4.87 |
| | # 5 | **4.83** | - | 5.77 |
| | # 6 | 5.18 | **8.19** | 5.13 |
| | # 7 | 1.78 | **1.81** | 2.21 |

TABLE X
ISNR (in Decibels) Computed for Experiments Performed With the "Testpat1" Image. For Each Degradation, the Best Value is Shown in Bold

| Testpat1 256 × 256 | No noise | | BSNR at 30dB | |
|---|---|---|---|---|
| **Blur** | **Without constraints** | **With constraints** | **Without constraints** | **With constraints** |
| # 1 | 16.91 | **17.99** | **8.43** | 8.32 |
| # 2 | 13.85 | **15.49** | 10.23 | **11.05** |
| # 3 | 16.70 | **17.14** | 8.24 | **8.50** |
| # 4 | **15.98** | - | **14.59** | - |
| # 5 | **14.11** | - | **13.63** | - |
| # 6 | 9.56 | **14.22** | **18.05** | 14.98 |
| # 7 | 9.53 | **9.94** | 5.77 | **6.51** |

in which

$$f(x) = \sqrt{\sum_{\theta \in \Theta} (d_\theta * x)^2}. \qquad (25)$$

Cost function (24) is quadratic on the filter $h$

$$C(x,h) = \frac{1}{2}\left\|\mathrm{vec}(y) - X\mathrm{vec}(h)\right\|_2^2 + \lambda R\left(f(x)\right) \qquad (26)$$

in which $X$ is a matrix, corresponding to the linear operation of convolving an image with $x$, and vec() is the operation that vectorizes a matrix lexicographically. Considering (26), the gradient of (24) with respect to $h$ is given by

$$\frac{\partial C(x,h)}{\partial \mathrm{vec}(h)} = X^T\left[\mathrm{vec}(y) - X\mathrm{vec}(h)\right] \qquad (27)$$

in which $X^T$ is the transpose of $X$. Matrices $X$ and $X^T$ were not explicitly computed. Instead, their corresponding operations were performed in the frequency domain

$$X\mathrm{vec}(w) = \mathrm{ifft2}\left[\mathrm{fft2}(x) .* \mathrm{fft2}(w)\right] \qquad (28)$$
$$X^T\mathrm{vec}(w) = \mathrm{ifft2}\left[\mathrm{conj}\left(\mathrm{fft2}(x)\right) .* \mathrm{fft2}(w)\right] \qquad (29)$$

in which fft2() is the 2-D discreet Fourier transform and ifft2() is its inverse. The notations $.*$ and $\mathrm{conj}()$ denote the point-wise product and complex conjugation operations, respectively.

Let $H$ and $D_\theta$ be the matrix operators corresponding to convolving an image with $h$ and $d_\theta$, respectively. Cost function (24) can also be written as

$$\frac{1}{2}\left\|\mathrm{vec}(y) - H\mathrm{vec}(x)\right\|_2^2 + \lambda R\left[f(x)\right] \tag{30}$$

in which

$$R\left(f(x)\right) = \sum_i \left[\left(\sqrt{\sum_{\theta\in\Theta}\left(D_\theta \mathrm{vec}(x)\right)^2}\right)_i + \epsilon\right]^q. \tag{31}$$

The derivative of (24) with respect to $x$ is then given by

$$\frac{\partial C(x,h)}{\partial \mathrm{vec}(x)} = H^T\left[\mathrm{vec}(y) - H\mathrm{vec}(x)\right] + \lambda\frac{\partial R\left[f(x)\right]}{\partial \mathrm{vec}(x)} \tag{32}$$

in which

$$\frac{\partial R\left[f(x)\right]}{\partial \mathrm{vec}(x)} = \sum_{\theta\in\Theta} D_\theta^T\left[.\frac{D_\theta x}{\mathrm{vec}\left[f(x)\right]}.*\ \mathrm{vec}\left[\left(f(x)+\epsilon\right)^{q-1}\right]\right)\Bigg] \tag{33}$$

and $.-.$ is the point-wise division operator. Again, the products by $H$, $H^T$, $D_\theta$ and $D_\theta^T$ can be computed in the frequency domain, similarly to (29).

## Acknowledgment

The authors would like to thank M. Kaveh and Y.-L. You for having provided the code of the BID method proposed in [2]. They would also like to thank to R. J. Plemmons and D. Chen for having provided the satellite image that we have used, as well as their colleagues J. Bioucas-Dias and M. Figueiredo for very useful discussions, and the reviewers for their comments, which significantly contributed to improve the quality of this paper.

## References

[1] Y. P. Guo, H. P. Lee, and C. L. Teo, "Blind restoration of images degraded by space-variant blurs using iterative algorithms for both blur identification and image restoration," *Image Vis. Comput.*, vol. 15, no. 5, pp. 399–410, 1997.

[2] Y.-L. You and M. Kaveh, "Blind image restoration by anisotropic regularization," *IEEE Trans. Image Process.*, vol. 8, no. 3, pp. 396–407, Mar. 1999.

[3] M. Blume, D. Zikic, W. Wein, and N. Navab, "A new and general method for blind shift-variant deconvolution of biomedical images," in *Proc. MICCAI (1)*, 2007, pp. 743–750.

[4] M. Sorel and J. Flusser, "Space-variant restoration of images degraded by camera motion blur," *IEEE Trans. Image Process.*, vol. 17, no. 1, pp. 105–116, Jan. 2008.

[5] M. Welk, D. Theis, and J. Weickert, "Variational deblurring of images with uncertain and spatially variant blurs," in *Proc. DAGM Symp.*, 2005, pp. 485–492.

[6] A. Kubota and K. Aizawa, "Reconstructing arbitrarily focused images from two differently focused images using linear filters," *IEEE Trans. Image Process.*, vol. 14, no. 11, pp. 1848–1859, Nov. 2005.

[7] L. Bar, B. Berkels, M. Rumpf, and G. Sapiro, "A variational framework for simultaneous motion estimation and restoration of motion-blurred video," in *Proc. IEEE 11th Int. Conf. Computer Vision*, 2007, pp. 1–8.

[8] M. S. C. Almeida and L. B. Almeida, "Blind deblurring of foreground-background images," presented at the Int. Conf. Image Processing, Cairo, Egypt, Nov. 2009.

[9] R. Fergus, B. Singh, A. Hertzmann, S. T. Roweis, and W. T. Freeman, "Removing camera shake from a single photograph," *ACM Trans. Graph. (TOG)*, vol. 25, pp. 787–794, 7 2006.

[10] M. Bertero and P. Boccacci, *Introduction to Inverse Problems in Imaging*. Bristol, U.K.: IOP, 1998.

[11] M. S. C. Almeida and L. B. Almeida, "Blind deblurring of natural images," in *Proc. ICASSP*, Las Vegas, NV, Mar. 2008, pp. 1261–1264.

[12] M. Bertero and P. Boccacci, "Image restoration methods for the large binocular telescope," *Astron. Astrophys. Suppl.*, vol. 147, pp. 323–333, 2000.

[13] J. P. Muller, *Digital Image Processing in Remote Sensing*. New York: Taylor & Francis, 1988.

[14] S. Jefferies, K. Schulze, C. Matson, K. Stoltenberg, and E. K. Hege, "Blind deconvolution in optical diffusion tomography," *Opt. Exp.*, vol. 10, pp. 46–53, 2002.

[15] A. M. Bronstein, M. M Bronstein, M. Zibulevsky, and Y. Y. Zeevi, "Quasi-maximum likelihood blind deconvolution of images acquired through scattering media," in *Proc. ISBI*, Apr. 2004, pp. 352–355, 4.

[16] J. Markham and J. Conchello, "Parametric blind deconvolution: A robust method for the simultaneous estimation of image and blur," *J. Opt. Soc. Amer. A*, pp. 2377–2391, 1999.

[17] D. Adam and O. Michailovich, "Blind deconvolution of ultrasound sequences using non-parametric local polynomial estimates of the pulse," *IEEE Trans. Biomed. Eng.*, vol. 42, no. 2, pp. 118–131, Feb. 2002.

[18] P. Pankajakshan, B. Zhang, L. Blanc-Feraud, Z. Kam, J.-C. Olivo-Marin, and J. Zerubia, "Blind deconvolution for diffraction-limited fluorescence microscopy," presented at the ISBI, 2008.

[19] G. K. Chantas, N. P. Galatsanos, and A. C. Likas, "Bayesian restoration using a new nonstationary edge-preserving image prior," *IEEE Trans. Image Process.*, vol. 15, no. 10, pp. 2987–2997, Oct. 2006.

[20] J. A. Guerrero-Colon and J. Portilla, "Deblurring-by-denoising using spatially adaptive gaussian scale mixtures in overcomplete pyramids," in *Proc. IEEE Int. Conf. Image Processing*, Atlanta, GA, Oct. 2006, pp. 625–628.

[21] L. Bar, N. Kiryati, and N. Sochen, "Image deblurring in the presence of impulsive noise," *Int. J. Comput. Vis.*, vol. 70, no. 3, pp. 279–298, Mar. 2006.

[22] V. Katkovnik, K. Egiazariana, and J. Astola, "A spatially adaptive nonparametric regression image deblurring," *IEEE Trans. Image Process.*, vol. 14, no. 10, pp. 1469–1478, Oct. 2005.

[23] G. Chantas, N. Galatsanos, A. Likas, and M. Saunders, "Variational bayesian image restoration based on a product of t-distributions image prior," *IEEE Trans. Image Process.*, vol. 17, no. 10, pp. 1795–805, Oct. 2008.

[24] D. Kundur and D. Hatzinakos, "Blind image deconvolution," *IEEE Sig. Process. Mag.*, pp. 43–64, May 1996.

[25] P. Campisi and K. Egiazarian, *Blind Image Deconvolution: Theory and Applications*. Boca Raton, FL: CRC, 2007.

[26] H. Yin and I. Hussain, "Blind source separation and genetic algorithm for image restoration," presented at the ICAST, Islamabad, Pakistan, Sep. 2006.

[27] F. Krahmer, Y. Lin, B. McAdoo, K. Ott, J. Wang, D. Widemannk, and B. Wohlberg, Blind Image Deconvolution: Motion Blur Estimation, Tech Rep., Univ. Minnesota, 2006.

[28] J. Oliveira, M. Figueiredo, and J. Bioucas-Dias, "Blind estimation of motion blur parameters for image deconvolution," presented at the Iberian Conf. Pattern Recognition and Image Analysis, Girona, Spain, Jun. 2007.

[29] S. Chang, S. W. Yun, and P. Park, "PSF search algorithm for dual-exposure typeblurred," *Int. J. Appl. Sci., Eng., Technol.*, vol. 4, no. 1, pp. 1307–4318, 2007.

[30] S. D. Babacan, R. Molina, and A. K. Katsaggelos, "Variational bayesian blind deconvolution using a total variation prior," *IEEE Trans. Image Process.*, vol. 18, no. 1, pp. 12–26, Jan. 2009.

[31] K.-H. Yap, L. Guan, and W. Liu, "A recursive soft-decision approach to blind image deconvolution," *IEEE Trans. Signal Process.*, vol. 51, no. 2, Feb. 2003.

[32] L. Justen and R. Ramlau, "A non-iterative regularization approach to blind deconvolution," *Inst. Phys. Pub. Inv. Probl.*, vol. 22, pp. 771–800, 2006.

[33] A. S. Carasso, “Direct blind deconvolution,” *SIAM J. Appl. Math.*, 2001.

[34] A. S. Carasso, “The APEX method in image sharpening and the use of low exponent Évy stable laws,” *SIAM J. Appl. Math.*, vol. 63, no. 2, pp. 593–618, 2002.

[35] R. Molina, J. Mateos, and A. K. Katsaggelos, “Blind deconvolution using a variational approach to parameter, image, and blur estimation,” *IEEE Trans. Image Process.*, vol. 15, no. 12, pp. 3715–3727, Dec. 2006.

[36] Y.-L. You and M. Kaveh, “A regularization approach to joint blur identification and image restoration,” *IEEE Trans. Image Process.*, vol. 5, no. 3, pp. 416–428, Mar. 1996.

[37] T. F. Chan and C.-K. Wong, “Total variation blind deconvolution,” *IEEE Trans. Image Process.*, vol. 7, no. 3, Mar. 1998.

[38] L. He, A. Marquina, and S. J. Osher, “Blind deconvolution using TV regularization and bregman iteration,” *Int. J. Imag. Syst. Technol.*, vol. 15, pp. 74–83, 2005.

[39] R. Kaftory and N. Sochen, “Variational blind deconvolution of multichannel images,” *Int. J. Imag. Syst. Technol.*, vol. 15, no. 1, pp. 55–63, 2005.

[40] G. Panci, P. Campisi, S. Colonnese, and G. Scarano, “Multi-channel blind image deconvolution using the bussgang algo-rithm: Spatial and multiresolution approaches,” *IEEE Trans. Image Process.*, vol. 12, no. 11, pp. 1324–1337, Nov. 2003.

[41] F. Sroubek and J. Flusser, “Multichannel blind deconvolution of spatially misaligned images,” *IEEE Trans. Image Process.*, vol. 14, no. 7, pp. 874–883, Jul. 2005.

[42] L. Yuan, J. Sun, L. Quan, and H.-Y. Shum, “Blurred/no-blurred image alignment using kernel sparseness prior,” presented at the ICCV, 2007.

[43] V. Katkovnik, D. Paliy, K. Egiazarian, and J. Astola, “Frequency domain blind deconvolution in multiframe imaging using anisotropic spatially-adaptive denoising,” presented at the EUSIPCO, Florence, Italy, Sep. 2006.

[44] L. Yuan, J. Sun, L. Quan, and H.-Y. Shum, “Image deblurring with blurred/noisy image pairs,” presented at the Int. Conf. Computer Graphics and Interactive Techniques, San Diego, CA, 2007.

[45] M. M. Bronstein, A. M. Bronstein, and M. Zibulevsky, “Blind deconvolution of images using optimal sparse representations,” *IEEE Trans. Image Process.*, vol. 14, no. 6, pp. 726–735, Jun. 2005.

[46] D. Li, R. M. Mersereau, and S. Simske, “Blind image deconvolution through support vector regression,” *IEEE Trans. Neural Netw.*, vol. 18, no. 3, May 2007.

[47] I. Aizenberg, D. V. Paliy, J. M. Zurada, and J. T. Astola, “Blur identification by multilayer neural network based on multivalued neurons,” *IEEE Trans. Neural Netw.*, vol. 19, no. 5, pp. 883–898, May 2008.

[48] L. Rudin, S. Osher, and E. Fatemi, “Nonlinear total variation based noise removal algorithms,” *Phys. D*, vol. 60, pp. 259–268, 1992.

[49] L. G. Shapiro and G. C. Stockman, *Computer Vision*. Englewood Cliffs, NJ: Prentice-Hall, 2001, ch. 5.

[50] L. G. Roberts, Machine Perception on Three-Dimensional Solids, Tech. Rep. Massachusetts Inst. Technol., Lexington Lab., 1963.

[51] M. S. Prewitt, “Object enhancement and extraction,” *Picture Process. Psychopictorics*, 1970.

[52] L. S. Davis, “A survey of edge detection techniques,” *Comput. Graph. Image Process.*, vol. 4, pp. 248–270, 1973.

[53] M. Nikolova, “Analysis of the recovery of edges in images and signals by minimizing nonconvex regularized least-squares,” *SIAM J. Multiscale Model. Simul.*, vol. 4, no. 3, pp. 960–991, 2005.

[54] L. Almeida, “Multilayer perceptrons,” in *Handbook of Neural Computation*, E. Fiesler and R. Beale, Eds. Bristol, U.K.: IOP, 1997 [Online]. Available: http://www.lx.it.pt/~lbalmeida/papers/AlmeidaHNC.pdf

[55] S. D. Babacan, R. Molina, and A. K. Katsaggelos, “Parameter estimation in TV image restoration using variational distribution approximation,” *IEEE Trans. Image Process.*, vol. 17, no. 3, pp. 326–339, Mar. 2008.

[56] P. C. Hansen, *The L-Curve and Its Use in the Numerical Treatment of Inverse Problems*, ser. Advances in Computational Bioengineering. Southampton, U.K.: WIT Press, 2001.



**Mariana S. C. Almeida** was born in Lisbon, Portugal, on May 30, 1982. She graduated in electrical engineering from the Instituto Superior Técnico (IST, the Engineering School of the Technical University of Lisbon) in 2005, where she is currently pursuing the Ph.D. degree in electrical engineering. Her Ph.D. program is in the field of blind image separation and deconvolution, under the supervision of Prof. L. B. Almeida, at the IT-Instituto de Telecomunicações (Telecommunications Institute), Lisbon.



**Luís B. Almeida** was born in Lisbon, Portugal, on April 15, 1950. He graduated in electrical engineering from the Instituto Superior Técnico, Lisbon, in 1972, and received the “Doutor” degree from the Techincal University of Lisbon in 1983, with a thesis on nonstationary modeling of speech.

Since 1972, he has been with the Instituto Superior Técnico (IST), Technical University of Lisbon, where he has been a full professor since 1995 in the areas of signal processing and machine learning. From 1984 to 2004, he was Head of the Neural Networks and Signal Processing Group of INESC-ID. From 2000–2003, he was Chair of INESC-ID. In 2005, he joined the Instituto de Telecomunicações (Telecommunications Institute). Currently, he is Chair of the Electrical and Computer Engineering Department, IST. He is the author of many papers on speech modeling and coding, time-frequency representations of signals and the fractional Fourier transform, learning algorithms for neural networks, independent component analysis/source separation, and image processing. Currently, his research focuses mainly on blind image deblurring.

Prof. Almeida was the recipient of an IEEE Signal Processing Area ASSP Senior Award and of several national awards.