

Fast High-Quality non-Blind Deconvolution Using Sparse Adaptive Priors

Horacio E. Fortunato · Manuel M. Oliveira

Received: date / Accepted: date

Abstract We present an efficient approach for high-quality non-blind deconvolution based on the use of sparse adaptive priors. Our regularization term enforces preservation of strong edges while removing noise. We model the image-prior deconvolution problem as a linear system, which is solved in the frequency domain. Our approach's clean formulation lends to a simple and efficient implementation. We demonstrate its effectiveness by performing an extensive comparison with existing non-blind deconvolution methods, and by using it to deblur actual photographs degraded by camera shake or motion. Our experiments show that our solution is faster and its results tend to have higher peak signal-to-noise ratio (PSNR) than the state-of-the-art techniques. Thus, it provides an attractive alternative to perform high-quality non-blind deconvolution of large images, as well as to be used as the final step of blind-deconvolution algorithms.

Keywords non-blind deconvolution, adaptive priors, deblurring, computational photography.

1 Introduction

Deconvolution is a fundamental operation in image processing, with applications spanning a large number of areas, ranging from microscopy [27] to astronomy [28]. In computational photography, deconvolution algorithms

are at the heart of several techniques, including depth estimation from single photographs [19,12], defocus deblurring [37], and motion deblurring due to camera shake [11,26,4,6] or object movement [25,20].

Image deconvolution tries to obtain a sharp image f having as input a blurred version g , and possibly a convolution kernel h . If h is available, the process is called *non-blind deconvolution*. Both blind and non-blind deconvolution are highly ill-posed problems, accepting a large or infinite number of solutions. Given its importance, image deconvolution has received significant attention from the image and signal processing communities [1,5]. Recently, several techniques exploring natural-image statistics to constrain the problem have been proposed [18,26,33,16]. They exploit the use of a sparse distribution of image derivatives as natural-image priors and achieve good results for non-blind image deconvolution.

We present an efficient approach for high-quality non-blind deconvolution that is faster and whose results tend to present higher PSNR than the ones obtained with state-of-the-art methods. Our solution is based on a regularization technique using sparse adaptive priors, and its clean formulation lends to a very simple implementation. A key aspect of our technique is *a formulation for the sparse image-prior deconvolution problem that can be expressed as a linear system* and, therefore, be efficiently solved. Our adaptive priors penalize small derivative values, which tend to be associated to noise, but preserves large derivatives associated to image borders. Due to its efficiency and high-quality results, our approach is an attractive alternative to perform non-blind deconvolution of large images, as well as to be used as the final step of recent blind-deconvolution algorithms, such as [11,26,6].

Horacio E. Fortunato
Uniritter - Laureate International Universities
Porto Alegre - RS, Brazil
E-mail: horacio_fortunato@uniritter.edu.br

Manuel M. Oliveira
Instituto de Informática, UFRGS
Porto Alegre - RS, Brazil
E-mail: oliveira@inf.ufrgs.br

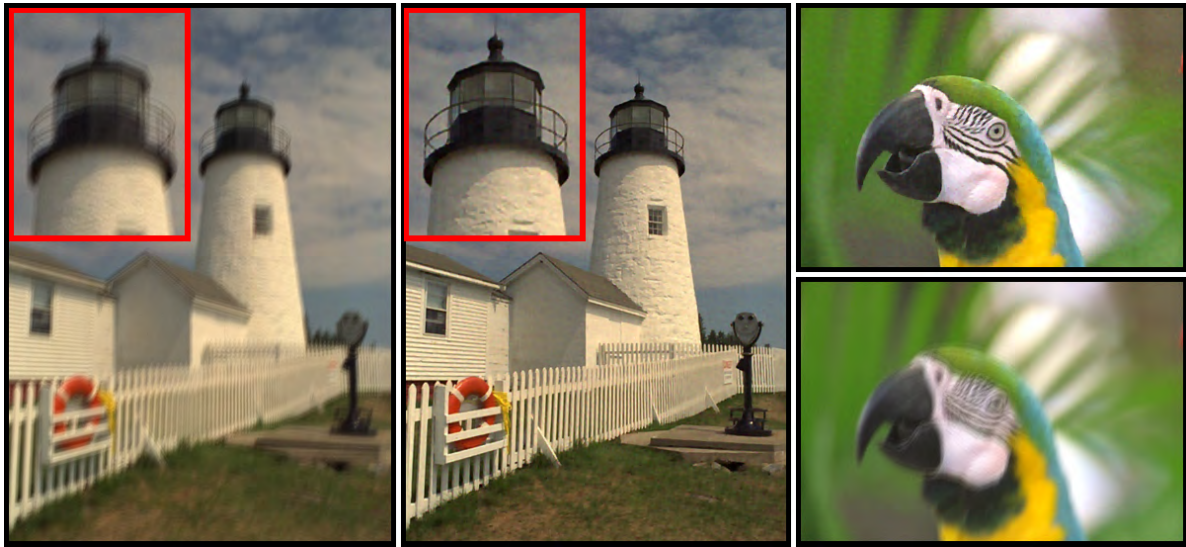


Fig. 1 Examples of non-blind deconvolution obtained with our method. The blurred images on the left (lighthouse) and bottom right (parrot) were created convolving two images from the Kodak Lossless True Color Image Suite with the 19×19 blur kernel of Krishnan and Fergus [16] and adding 1% of Gaussian noise. The results recovered by our method are shown on the center and on the top right, respectively. Note the proper reconstruction of fine image details. For these color images (768×512 pixels) our method takes approximately 0.82 seconds using a MATLAB implementation on a 3.4 GHz i7 CPU.

We demonstrate the effectiveness of our approach by performing extensive comparisons against existing methods, and by using our technique to deblur actual photographs degraded by camera shake or motion. Fig. 1 shows examples of deconvolved images obtained with our technique. Note how fine details in the lighthouse’s handrail and on the parrot’s head are properly reconstructed. They illustrate the ability of our technique to produce high-quality deconvolutions.

The **contributions** of our work include:

- An efficient approach to perform high-quality non-blind deconvolution based on adaptive-prior regularization (Section 4). Our approach is faster and its results tend to present higher PSNR than state-of-the-art methods. It also has a simple implementation;
- A formulation for the sparse image-prior deconvolution problem that is expressed as a linear system and efficiently solved in frequency domain (Section 4);
- A padding technique to allow deblurring to be performed using frequency domain deconvolution without introducing border ringing artifacts (Section 4.1).

2 Related Work

There is a substantial amount of literature on image deconvolution. Here, we discuss the non-blind image-deconvolution techniques that are closely related to ours. For a review of blind-deconvolution methods we refer

the readers to [5], whereas a discussion of classical non-blind-deconvolution algorithms can be found in [1].

Recently, several techniques exploring natural-image statistics as image priors to constrain the deconvolution problem have been proposed [18, 26, 33, 14, 16]. Levin et al. [18] discuss the use of both Gaussian and hyper-Laplacian priors. Gaussian priors lend to a linear system that can be efficiently solved, but tend to introduce excessive blurring and ringing artifacts. Better results are achieved with hyper-Laplacian priors, which require solving a non-convex optimization problem. Levin et al. approximate its solution using iterative reweighted least squares (IRLS) [29], a process typically taking hundreds of iterations to converge.

Shan et al. [26] separate convolution from other operations, performing the former in frequency domain. Their technique is faster than Levin et al.’s IRLS approach, but still about ten times slower than ours.

Yuan et al. [34] use a multi-scale approach for image deconvolution intended for reducing the occurrence of ringing artifacts. The inter-scale step of the algorithm combines Lucy-Richardson deconvolution [23] with bilateral filtering [32]. At each scale, an iterative residual deconvolution is performed. While the method succeeds in reducing rings, it suffers from the same sensitivity to noise as Lucy-Richardson, and requires a large number of parameters (ten).

Joshi et al. [14] use local color statistics as priors for deblurring and denoising. The technique uses hyper-Laplacian priors, and the solution of the resulting non-

convex optimization is also approximated using IRLS, which affects its performance.

Two related techniques were presented by Wang et al. [33] and by Krishnan and Fergus [16]. Both methods exploit half-quadratic minimization (HQM) [13] for acceleration. HQM is a technique that replaces a non-convex minimization problem with a new one that has the same global minimum, but is easier to compute. This new problem can be split into two parts containing decoupled variables, which are jointly optimized using alternating phases. Wang et al. [33] showed how HQM can be applied to image deconvolution using total-variation (TV) regularization. A variant of their method using a Laplacian (ℓ_1) prior is also available.

The technique by Krishnan and Fergus [16] uses a hyper-Laplacian prior. Their second phase of the HQM requires estimating the values of two variables, for which they present two alternative solutions: an analytical (restricted to the hyper-Laplacian values of $\alpha = 1/2$ and $\alpha = 2/3$), and one based on a lookup table (LUT) containing 2.56-million entries. The LUT approach is preferred, as it is about 5 times faster than the analytic one, while supporting any value of α (by interpolation).

Cho et al. [8] try to enforce that the reconstructed image has a gradient distribution similar to the reference one. Their approach tends to better reproduce mid-frequency textures, but the resulting PSNRs are lower than those obtained using other methods.

In contrast to all previous approaches, our technique is based on a sparse adaptive-prior regularization and solved as a linear system. Our priors are adaptive in the sense that they are based on estimates of the image's derivatives. We present a comprehensive comparison of non-blind deconvolution techniques and show that our method is faster and our results tend to achieve higher PSNR than previous solutions.

3 Image Capture and Deconvolution

The image-capture process is traditionally modeled as the convolution of a blur kernel $\mathbf{h}$ with an *ideal* sharp image $\mathbf{f}$, plus some noise $\mathbf{n}$:

$$\mathbf{g} = \mathbf{h} \otimes \mathbf{f} + \mathbf{n}. \quad (1)$$

Assuming $\mathbf{n}$ is a Gaussian white-noise process consisting of independent random variables, $\mathbf{g}$ is the realization of a random array with probability distribution determined by the ideal image $\mathbf{f}$ and kernel $\mathbf{h}$. The random variables $\mathbf{g}_{i,j}$ are independent and normally distributed.

Given the ill-posed nature of the problem represented by Eq. (1), it can be stably solved using regu-

larization methods [31, 9]:

$$\hat{\mathbf{f}} = \arg \min_{\mathbf{f}} (\delta_{DATA}(\mathbf{f}) + \lambda \delta_{REG}(\mathbf{f})). \quad (2)$$

Here, $\hat{\mathbf{f}}$ is an estimate of $\mathbf{f}$. The first term in Eq. (2) enforces faithfulness to the captured image $\mathbf{g}$ (data), while the regularization term adds some penalty based on constraints imposed on $\mathbf{f}$. λ defines the relative weights for the two terms. For the data term, the negative log-likelihood (*i.e.*, $-\log p(\mathbf{f}|\mathbf{g}, \mathbf{h})$) is the natural choice. Thus, given the image degradation model with white Gaussian noise of Eq. (1), the data term is often modeled as $\delta_{DATA}(\mathbf{f}) = \|\mathbf{h} \otimes \mathbf{f} - \mathbf{g}\|_2^2$, where $\|\cdot\|_2$ is the ℓ_2 norm. If the regularization term is also quadratic, Eq. (2) can be evaluated as a linear system obtained from the corresponding normal equation.

4 Efficient non-Blind Deconvolution

Our goal is to obtain a sharp image $\hat{\mathbf{f}}$ that *explains* the acquired image $\mathbf{g}$ under convolution with kernel $\mathbf{h}$, such that $\hat{\mathbf{f}}$ qualifies as a *natural* image. To avoid excessive blurring and remove noise, *we formulate our prior in such a way that it preserves strong edges, while penalizing ones below a given threshold* (noise level). We express Eq. (1) in matrix form as:

$$\mathbf{g} = \mathbf{h} \mathbf{f} + \mathbf{n}, \quad (3)$$

where $\mathbf{h}$ is a square matrix representing the linear convolution operation with $\mathbf{h}$. Terms $\mathbf{g}$, $\mathbf{f}$, and $\mathbf{n}$ are vectorized versions of $\mathbf{g}$, $\mathbf{f}$, and $\mathbf{n}$, respectively (Eq. (1)). Assuming that the captured image $\mathbf{g}$ has R rows and C columns, $\mathbf{h}$ is an $(R \times C)$ -square matrix, and $\mathbf{f}$, $\mathbf{g}$ and $\mathbf{n}$ are column vectors with $(R \times C)$ elements. Our approach then computes $\hat{\mathbf{f}}$ as:

$$\hat{\mathbf{f}} = \arg \min_{\mathbf{f}} \delta(\mathbf{f}), \quad (4)$$

$$\delta(\mathbf{f}) = \|\mathbf{h} \mathbf{f} - \mathbf{g}\|_2^2 + \sum_{s=1}^5 \lambda_s \|\mathbf{d}_s \mathbf{f} - \mathbf{w}_s\|_2^2, \quad (5)$$

where $\mathbf{f}$, $\mathbf{h}$ and $\mathbf{g}$ have the same meaning as in Eq. (3). The matrices $\mathbf{d}_s$, $s \in \{1, \dots, 5\}$, represent the first and second-order-derivative filter operators: d_x, d_y, d_{xx}, d_{yy} and d_{xy} , respectively. λ_s are positive weights, and $\mathbf{w}_s$ are the *expected or specified responses* of these filters for the ideal image $\mathbf{f}$ (*i.e.*, $\mathbf{w}_s = \mathbf{d}_s \mathbf{f}$). The use of $\mathbf{w}_s$ in Eq. (5) allows us to specify a set of priors on the derivatives of $\mathbf{f}$. Thus, let τ be a threshold representing some noise level, and let $df_{i,j} = (\mathbf{d}_s \mathbf{f})_{i,j}$ be some derivative of image $\mathbf{f}$ at pixel coordinates (i, j) . For such a pixel, we let $\mathbf{w}_s = \mathbf{0}$ if $|df_{i,j}| < \tau$; otherwise, $\mathbf{w}_s = df_{i,j}$. Thus, *Eq. (5) selectively penalizes the occurrence of pixels with small derivatives (considered as*

noise), while preserving strong edges. We solve Eq. (4) by differentiating Eq. (5) with respect to each desired pixel f_m , and requiring the resulting expressions to be zero. This produces the following linear system, where $\square^T$ indicates matrix transpose:

$$(\mathbf{h}^T \mathbf{h} + \sum_{s=1}^5 \lambda_s d_s^T d_s) \hat{f} = \mathbf{h}^T \mathbf{g} + \sum_{s=1}^5 \lambda_s d_s^T w_s. \quad (6)$$

Eq. (6) can be rewritten as

$$\mathbf{a} \hat{f} = \mathbf{b}, \quad (7)$$

where

$$\begin{aligned} \mathbf{a} &= \mathbf{h}^T \mathbf{h} + \sum_{s=1}^5 \lambda_s d_s^T d_s, \\ \mathbf{b} &= \mathbf{h}^T \mathbf{g} + \sum_{s=1}^5 \lambda_s d_s^T w_s. \end{aligned} \quad (8)$$

The square matrix $\mathbf{a}$ is a convolution operator, as it is the sum of products of convolution operators. Eq. (7) can be expressed in the frequency domain as

$$\mathbf{A} \circ \hat{F} = \mathbf{B}, \quad (9)$$

where

$$\mathbf{A} = \mathbf{H}^* \circ \mathbf{H} + \sum_{s=1}^5 \lambda_s D_s^* \circ D_s, \quad (10a)$$

$$\mathbf{B} = \mathbf{H}^* \circ \mathbf{G} + \sum_{s=1}^5 \lambda_s D_s^* \circ W_s. \quad (10b)$$

Here, $\square^*$ represents complex conjugate, and $\circ$ is the element-wise matrix-product operator. $\mathbf{B} = \mathcal{F}(\mathbf{b})$, $\mathbf{H} = \mathcal{F}(\mathbf{h})$, $\mathbf{G} = \mathcal{F}(\mathbf{g})$, $D_s = \mathcal{F}(d_s)$, and $W_s = \mathcal{F}(w_s)$ are the frequency domain counterparts of $\mathbf{b}$, $\mathbf{h}$, $\mathbf{g}$, d_s , and w_s , respectively. $\mathcal{F}(\cdot)$ is the Fourier transform operator. $\hat{f}$ can then be obtained as

$$\hat{f} = \mathcal{F}^{-1}(\mathbf{B} ./ \mathbf{A}), \quad (11)$$

where $./$ stands for element-wise matrix division. Now, it is instructive to analyze some properties of Eq. (5):

- If all w_s coefficients are set to zero, it reduces to a standard Tikhonov regularization (*i.e.*, the Gaussian-prior formulation of Levin et al. [18]), which introduces ringings and other distracting artifacts;
- On the other extreme, if one could know in advance the derivatives of f and use them as w_s , Eq. (11) would give a very good approximation to f . The quality of such an approximation is limited by the standard deviation of the noise added to g (Eq. (1)).

Thus, it is key to our approach to *obtain an accurate estimate of w_s* , the derivatives of the ideal image f , which, of course, is not available. We solve this problem using a four-step process, which is illustrated in Fig. 3 and Algorithm 1:

Algorithm 1: Our non-Blind Deconvolution

Data: $\mathbf{g}$: captured image, $\mathbf{h}$: blurring kernel,
 λ_s : regularization weights
Result: $\hat{f}$: deblurred image
Step 1: *Gaussian deconvolution*
 Evaluate $\hat{f}^{(0)}$ using Eq. (10a, 10b and 11) with $W_s = 0$;
Step 2: *Edge-preserving smoothing filter*
 $\hat{f}^{(1)} \leftarrow \mathcal{EPS}(\hat{f}^{(0)})$; // $\mathcal{EPS}$: edge-preserving filter
Step 3: *Evaluate priors*
 Compute w_s using $\hat{f}^{(1)}$ and Eq. (12);
Step 4: *Actual deconvolution*
 Evaluate $\hat{f}$ using Eq. (10a, 10b and 11) with $W_s = \mathcal{F}(w_s)$; // $\mathcal{F}$: Fourier transform

- Step 1:** We obtain an initial approximation $\hat{f}^{(0)}$ to f by deconvolving the blurred image g using standard Tikhonov regularization (*i.e.*, $w_s = 0$). We call this step *Gaussian deconvolution* in Algorithm 1. Although $\hat{f}^{(0)}$ suffers from ringings and noise, it provides a good estimate for the edges of f (Fig. 3);
- Step 2:** A new estimate $\hat{f}^{(1)}$ is obtained by applying an edge-preserving smoothing filter to $\hat{f}^{(0)}$ to reduce noise while preserving important edges (Fig. 3);
- Step 3:** We compute the actual regularization priors w_s (Eq. 5) as a set of sparse first and second-order derivatives of $\hat{f}^{(1)}$ using Eq. (12) (Fig. 3);
- Step 4:** The deconvolved image $\hat{f}$ is finally obtained from g and w_s using Eqs. 10a, 10b, 11 (Fig. 3).

The regularization priors w_s are computed as

$$w_s = \Phi(d_s \hat{f}^{(1)}) = \frac{d_s \hat{f}^{(1)}}{(\frac{\tau}{d_s \hat{f}^{(1)}})^4 + 1}, \quad (12)$$

where $d_s = \{d_x, d_y, d_{xx}, d_{yy}, d_{xy}\}$, and the exponentiation $(\frac{\tau}{d_s \hat{f}^{(1)}})^4$ is element-wise. Eq. (12) *gradually zeroes the w_s prior values corresponding to pixels in $\hat{f}^{(1)}$ for which $|d_s \hat{f}^{(1)}| < \tau$* (see Fig. 4), thus penalizing their occurrence in Eq. (5). Such a gradual transition is required as, according to our experience, deconvolution results obtained by abruptly zeroing derivative values near τ tend to achieve lower PSNR when compared to the use of Eq. (12). There is a significant PSNR improvement from step 2 to step 4. For the example of Fig. 2, $\text{PSNR}(\hat{f}^{(1)}) = 32.33$ while $\text{PSNR}(\hat{f}) = 33.19$.

4.1 Removing Border Ringing Artifacts

Frequency-domain deconvolution produces correct results when applied to images created through circular convolution. However, the blurring introduced by the image-capture process (Eq. (1)) does not correspond to circular convolution. As a result, frequency-domain



Fig. 2 Example of image deblurring using our technique. (top) Blurry image g obtained by convolving the original sharp image with the 19×19 blur kernel of Krishnan and Fergus [16] and adding 1% of Gaussian noise. (bottom) Resulting deblurred image $\hat{f}$. Fig. 3 illustrates the steps of Algorithm 1 with the highlighted portion of the yellow cap.

deconvolution tends to introduce ringing artifacts at the borders of the reconstructed images [26, 33, 16]. To minimize the impact of these artifacts, Krishnan and Fergus [16] use edge-tapering operations, and Shan et al. [26] use a technique by Liu and Jia [22]. Since the problem is restricted to image borders, we use a simpler, but effective solution: pad the input image before performing deconvolution, and crop its result to remove the extra pixels (Fig. 5). Padding is done by replicating the image's first and last columns and rows k times. The minimum value for k should vary with the size of the kernel h . In all of our experiments, we use $k = 2m$, where m is the maximum between the kernel's width and height. We then multiply the padded image by a fading mask that gradually transitions from one, over the original image region, to almost zero towards the padded image borders. After deconvolution, we reverse the process by dividing the results by the mask and crop the image, thus removing the padding. The fading mask has the same dimensions as the padded image, and resembles a low-pass Butterworth filter with the

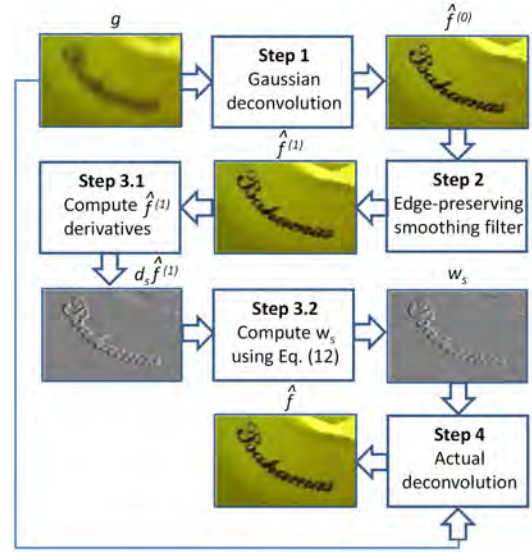


Fig. 3 Step-by-step illustration of Algorithm 1 for the highlighted portion of the yellow cap shown in Fig. 2 (top). Step 1: initial approximation $\hat{f}^{(0)}$ to f obtained by deconvolving the blurred image g using $w_s = 0$ (Gaussian deconvolution). Step 2: new estimate $\hat{f}^{(1)}$ obtained applying an edge-preserving smoothing filter to $\hat{f}^{(0)}$ to reduce noise while preserving important edges. Step 3: compute regularization priors w_s from the derivatives of $\hat{f}^{(1)}$; Step 4: the actual deconvolved image $\hat{f}$ is obtained from g and w_s using Eq. (11).

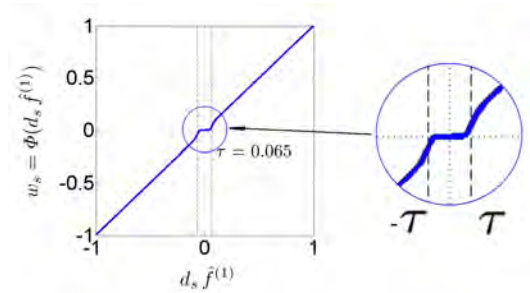


Fig. 4 Plot of Eq. (12) for a threshold $\tau = 0.065$. Values of the first and second order derivatives $d_s \hat{f}^{(1)}$ of $\hat{f}^{(1)}$, whose absolute values are less than τ are gradually set to zero. An example of its use is shown in Fig. 3, Step 3.2.

cutoff over the image borders:

$$\text{mask}(r, c) = \frac{1}{(1 + (\frac{r-r_c}{R/2})^{2n_r})(1 + (\frac{c-c_c}{C/2})^{2n_c})}, \quad (13)$$

where r and c are the row and column of a pixel in the mask, r_c and c_c are the mask's central row and column, R and C are the dimensions of the unpadded image, and n_r and n_c are selected to ensure a soft transition from 1 (over the image) to almost zero at the padded image borders: $n_r = \lceil 0.5 \log((1 - \alpha)/\alpha) / \log(r_c/(R/2)) \rceil$ and $n_c = \lceil 0.5 \log((1 - \alpha)/\alpha) / \log(c_c/(C/2)) \rceil$, where α is the desired value at the mask borders. In our exper-

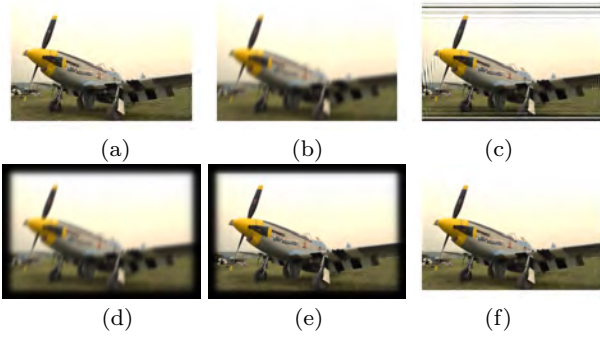


Fig. 5 Avoiding ringing artifacts at image borders when performing deconvolution in frequency domain. (a) Original image. (b) Blurred image obtained by convolving the image in (a) with the 19×19 blur kernel of Krishnan and Fergus [16] and adding 1% of Gaussian noise. (c) Deconvolved image using our method without padding. Note the ringing artifacts due to frequency-domain deconvolution. (d) Blurred image padded and faded out on the borders. (e) The resulting deconvolved image without ringing artifacts (before unpadding). (f) Final deconvolved image without ringing artifacts obtained from (e) after unpadding.

iments we used $\alpha = 0.01$. The entire process is illustrated in Fig. 5.

5 Results

We have implemented the described technique using MATLAB and used it to deconvolve a large number of images. This section describes some of these experiments, and provides an extensive comparison with eight non-blind image-deconvolution techniques. They include classical deconvolution algorithms, such as Lucy-Richardson (*Lucy*) and the generalized Wiener deconvolution method by Zhou et al. [37] (*Zhou*); Levin et al.’s Gaussian prior (ℓ_2) deconvolution, plus several sparse-image-prior solutions: Levin et al.’s sparse prior (ℓ_1) deconvolution (*IRLS* $\alpha = 4/5$) [18], Wang et al.’s [33] total variation (*TV*) and L_1 (ℓ_1), Shan et al.’s [26] technique (*Shan*), and Krishnan and Fergus’s LUT-based approach for $\alpha = 2/3$ (*Krishnan LUT* $\alpha = 2/3$) [16]. All comparisons were performed using software provided in the authors’ websites. For each technique, we used parameter values that produce best quality results. For Levin et al.’s hyper-Laplacian method, we used their implementation that includes second-order derivatives and 200 iterations, as it achieved higher PSNR values. All measurements use MATLAB R2010b on a 3.4 GHz i7 CPU, 16 GB of RAM, and Windows 7 64-bits.

We compare the various deconvolution methods using the *Kodak Lossless True Color Image Suite* from PhotoCD PCD0992 [15], a dataset of 24 images (768×512 or 512×768 pixels). The input images were blurred

using the 19×19 blur kernel of Krishnan and Fergus [16] and received 1% Gaussian white noise. Since these are color images, deconvolution is performed independently for each color channel. Examples of deconvolved results from this dataset obtained with our technique are shown in Figs. 1 and 2. Note how fine details in the lighthouse and in the parrot’s head in Fig. 1 are properly reconstructed. Similar comments can be made about the caps in Fig. 2. These results illustrate the effectiveness of our technique to produce high-quality deconvolutions.

Fig. 7 illustrates the results of the nine deconvolution algorithms applied to the image shown in Fig. 6 (right). These results show that some techniques are more sensitive to noise: Lucy-Richardson, Zhou et al. [37], Levin et al.’s Gaussian prior (ℓ_2) [18], and Wang et al.’s [33] ℓ_1 and TV. Other techniques exhibited higher tolerance to noise and produced cleaner reconstructions: Levin et al.’s hyper-Laplacian prior (IRLS), Shan et al.’s [26], Krishnan and Fergus’s [16], and our adaptive-prior regularization method. Fig. 8 compares the histograms of the first-order horizontal derivatives for some of the results shown in Fig. 7 against the corresponding histogram of the original image (Fig. 6 (left)). The comparison includes a noisy result (Zhou et al.) and three cleaner ones (Krishnan and Fergus’s, Levin et al.’s hyper-Laplacian prior (IRLS), and ours). Note how the cleaner ones closely match the distribution of small gradients of the original image.



Fig. 6 (left) Original image. (right) Blurred image used as input for the experiment in Fig. 7. It was obtained by convolving the image on the left with the 19×19 blur kernel of Krishnan and Fergus [16] and adding 1% of Gaussian noise.

As a quality metric for the deconvolved images, we use peak signal-to-noise ratio: $\text{PSNR} = 10 \log_{10} \frac{I_{max}^2}{MSE}$, where I_{max} is the maximum signal extent (e.g., 255 for eight-bit images), and MSE (mean-square error) is evaluated as $MSE = \frac{1}{n} \sum_x [\hat{I}(x) - I(x)]^2$. $\hat{I}(x)$ is the deconvolved image and $I(x)$ is the ground truth. For color images with R rows and C columns, the summation includes the three color channels, so $n = 3 \times R \times C$.

Table 1 summarizes the results of the experiment involving the Kodak Lossless True Color Image Suite. Our method achieves higher or same PSNR for 18 out



Fig. 7 Comparison of nine non-blind deconvolution techniques applied to the image shown in Fig. 6 (right). The first five techniques tend to produce noisier results, while the other ones exhibit higher tolerance to noise.

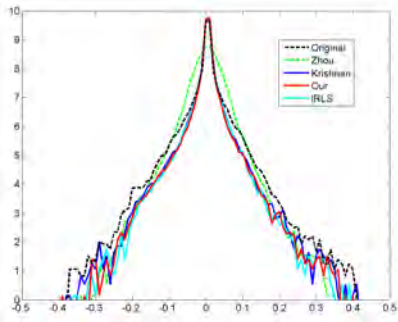


Fig. 8 Histogram of the first-order horizontal derivatives for some of the deconvolved images shown in Fig. 7, compared with the original one (Fig. 6 (left)). Our results reproduce the sparse derivative distribution of the original image.

of the 24 images (*i.e.*, 75%), against 2 for Levin et al.'s IRLS, 3 for Krishnan and Fergus's technique, and 1 for Wang et al.'s total variation (TV), while being faster than all of them (on average, $2.4\times$ faster than Krishnan and Fergus's, up to $764\times$ faster than IRLS). PSNR gain

is calculated as the difference between the PSNR of the deblurred image and the PSNR of the blurry image.

Table 2 shows the running times (in seconds) for the four deconvolution techniques most robust to noise (Levin et al.'s IRLS, Shan et al.'s, Krishnan and Fergus's, and ours) applied to five color images whose sizes vary from 256×256 to $3,072 \times 3,072$. The input images were blurred by a 13×13 blur kernel. Our technique is the fastest, with running times ranging from 0.16 sec for the 256^2 image, to 14.7 sec for the $3,072^2$ one.

Table 3 compares the PSNR and running-time results for the nine non-blind deconvolution techniques applied to a 768×512 color image (Fig. 2 original color image) blurred with eight different kernel sizes, ranging from 13×13 up to 41×41 , plus 1% Gaussian noise. Our method achieves higher PSNR for seven of the eight kernel sizes. It is also faster than all methods, except for ℓ_2 , but ours achieves significantly higher PSNR.

5.1 Camera-Motion Deblurring

We have also used our technique to deblur actual photographs affected by camera motion. Examples of these experiments are presented in Figs. 9 and 10, which show two blurry photographs with the corresponding recovered kernels (shown as insets at their bottom left), provided by Shan et al. [26]. Fig. 9 (left) shows two leaves, whose edges and stem are barely visible. The image on the right shows the deblurred result obtained with our technique. Note how the edges as well as the spots on the leaves became clearly visible. It is even possible to identify some darker plant structures (closer to the brick wall), which could not be perceived before.

Fig. 10 (left) shows a blurry tree. The deblurred result obtained with our technique is shown on the right. Our deconvolved images are qualitatively similar to the ones shown in [26]. Due to the lack of a ground truth, however, an objective comparison between our results and Shan et al.'s is not possible for these examples. We encourage the reader to perform a visual inspection of the corresponding images.



Fig. 9 Example of camera-motion deblurring. (left) Blurry photograph with corresponding camera-motion kernel (shown as an inset) provided by Shan et al.'s [26]. (right) Deblurred result produced by our technique. Note the clearly visible edges and spots on the leaves.



Fig. 10 Another example of camera-motion deblurring. (left) Input photograph and camera-motion kernel pair (by Shan et al [26]). (right) Recovered image obtained with our technique. Note the considerable improvement in sharpness.

5.2 Parameter Determination

The regularization parameters λ_s control the weight of each derivative in the regularization term of Eq. (5). For the experiments with the Kodak Image Suit reported in Table 1, we used the same value of λ_s for all values of s (all first and second-order derivatives). We found that $\lambda = 0.001$ for the Gaussian-deconvolution step (Step 1 of Algorithm 1 and Fig. 3) and $\lambda = 0.05$ for the actual deconvolution step (Step 4 of Algorithm 1 and Fig. 3) gave the best PSNR results. These values may need to be adjusted for other noise levels and deconvolution kernels. As for the edge-preserving smoothing filter, we used the domain-transform-based RF filter by Gastal and Oliveira [10], due to its computational efficiency and high-quality results. It uses two parameters: σ_s and σ_r , the spatial and range standard deviation, respectively. We used $\sigma_s = 20$ and $\sigma_r = 0.033$.

For the camera-motion-deblurring example shown in Fig. 9, we used the same parameters used for the Kodak Suit. For Tree example in Fig. 10, we used slightly different parameter values: $\lambda = 0.0001$ for the Gaussian-deconvolution step, $\lambda = 0.01$ for the actual deconvolution step, $\sigma_s = 20$, and $\sigma_r = 0.01$.

Finally, to compute our sparse priors w_s (Eq. (12)) for all experiments, we used $\tau = 0.065$ for first-order derivatives and $\tau = 0.0325$ for second-order ones. The supplementary materials include the MATLAB code and scripts used in our experiments (Tables 1-3).

5.3 Discussion and Limitations

Like other deblurring techniques, ours assumes a linear blurring model with Gaussian white noise (Eq. (1)). In natural images, the presence of saturated regions or pixels with uncorrelated values tend to introduce artifacts in the deconvolved images that are common to all deconvolution methods. A possible treatment to this problem is presented by Cho et al. [7]. Another deviation of the linear model results from the nonlinearity of the camera response function (CRF). It is usual to undo the nonlinearity by assuming a gamma function or to use the raw output of the digital camera. A detailed discussion of this subject can be found in [30].

PSNR values obtained for RGB images are slightly higher than for grayscale ones. This is because the edge-preserving smoothing filter that we use computes the distance between adjacent pixels using the ℓ_1 norm in XYRGB space. This distance is used to filter the three channels, taking advantage of the inter-channel coherence to reduce noise.

Image size	IRLS $\alpha = 4/5$	Shan	Krishnan LUT $\alpha = 2/3$	Our
256×256	39.46	1.23	1.01	0.16
512×512	154.40	5.70	1.36	0.55
1024×1024	626.75	17.48	3.27	1.61
2048×2048	2871.22	-	13.18	6.66
3072×3072	5875.10	-	27.98	14.07

Table 2 Running times (seconds) for the four deconvolution techniques most robust to noise applied to color images of different sizes convolved with a 13×13 blur kernel. Shan’s code reported an error for the 2048^2 and 3072^2 images.

6 Conclusion

We have presented an efficient approach for high-quality non-blind deconvolution based on the use of sparse adaptive priors. We model the problem as a linear system and solve it in the frequency domain. Our approach’s clean formulation lends to a simple and efficient implementation. We have performed extensive comparisons of our technique with eight non-blind deconvolution methods, which included image-quality assessment using PSNR, and running time. We have compared the performance of the techniques considering various image sizes, as well as blur-kernel sizes. We have also used our technique to deblur photographs affected by camera motion. These experiments demonstrate the effectiveness of our solution, showing that it produces higher PSNR and is faster than all evaluated noise-tolerant techniques. Given such desirable properties, our technique provides an attractive alternative to perform high-quality non-blind deconvolution of large images, as well as to be used as the final step of blind-deconvolution algorithms [11, 26, 6].

Acknowledgements This work was sponsored by CNPq (grants 482271/2012-4 and 308936/2010-8). We thank the authors of the compared techniques for making their code available, and Shan et al. for providing the blurry photographs and camera-motion kernels shown in Figs. 9 and 10. The images used for technique comparisons are from the *Kodak Lossless True Color Image Suite* – PhotoCD PCD0992 [15].

References

- BANHAM, M. AND KATSAGGELOS, A. 1997. Digital image restoration. *Signal Processing Magazine, IEEE* 14,2,4-41.
- BEN-EZRA, M., AND NAYAR, S. 2004. Motion-based Motion Deblurring. *IEEE TPAMI* 26, 6 (Jun), 689-698.
- BONESKY, T. 2009. Morozov’s discrepancy principle and tikhonov-type functionals. *Inverse Problems* 25, 1, 015015.
- CAI, J.-F., JI, H., LIU, C., AND SHEN, Z. 2009. Blind motion deblurring from a single image using sparse approximation. In *CVPR 2009*, 104–111.

- CAMPISI, P., AND EGAZARIAN, K. 2007. *Blind Image Deconvolution: Theory and Applications*. CRC Press.
- CHO, S., AND LEE, S. 2009. Fast motion deblurring. *ACM Trans. Graph.* 28, 5 (Dec.), 145:1–145:8.
- CHO, S., AND WANG J. AND LEE, S. 2011. Handling Outliers in Non-blind Image Deconvolution. *ICCV 2011*, 1-8.
- CHO, T.S. AND ZITNICK, C.L. AND JOSHI, N. AND KANG, S.B. AND SZELISKI, R. AND FREEMAN W.T. 2012. Image Restoration by Matching Gradient Distributions. *IEEE TPAMI*. 34, 4, 683-694.
- ENGL, H., HANKE, M., AND NEUBAUER, A. 1996. *Regularization of Inverse Problems*. Kluwer.
- GASTAL, E. S. L. AND OLIVEIRA, M. M. 2011. *Domain Transform for Edge-Aware Image and Video Processing*. *ACM TOG*. 30, 4, 69:1-12. SIGGRAPH 2011.
- FERGUS, R., SINGH, B., HERTZMANN, A., ROWEIS, S. T., AND FREEMAN, W. T. 2006. Removing camera shake from a single photograph. *ACM TOG* 25 (July), 787–794.
- FORTUNATO, H. E., AND OLIVEIRA, M. M. 2012. Coding depth through mask structure. *Computer Graphics Forum* 31, 2 (May), 459–468. Proceedings of Eurographics.
- GEMAN, D., AND YANG, C. 1995. Nonlinear image recovery with half-quadratic regularization. *Image Processing, IEEE Transactions on* 4, 7, 932–946.
- JOSHI, N., ZITNICK, C., SZELISKI, R., AND KRIEGMAN, D. 2009. Image deblurring and denoising using color priors. In *CVPR 2009*, 1550–1557.
- KODAK, 2013. Kodak Lossless True Color Image Suite. <http://r0k.us/graphics/kodak/>. Last access, Jan. 2013.
- KRISHNAN, D., AND FERGUS, R. 2009. Fast image deconvolution using hyper-laplacian priors. In *Advances in Neural Information Processing Systems* 22, 1033-1041.
- KRISHNAN, D., AND FERGUS, R., 2013. Fast image deconvolution using hyper-laplacian priors - website. <http://cs.nyu.edu/dilip/research/fast-deconvolution/>.
- LEVIN, A., FERGUS, R., DURAND, F., AND FREEMAN, W. T. 2007. Deconvolution using natural image priors. *ACM Trans. Graphics* 26, 3, 0–2. Suppl. materials.
- LEVIN, A., FERGUS, R., DURAND, F., AND FREEMAN, W. T. 2007. Image and depth from a conventional camera with a coded aperture. *ACM TOG*. 26 (July). Article 70.
- LEVIN, A., SAND, P., CHO, T. S., DURAND, F., AND FREEMAN, W. T. 2008. Motion-invariant photography. *ACM Trans. Graph.* 27 (August), 71:1–71:9.
- LEVIN, A., WEISS, Y., DURAND, F., AND FREEMAN, W. 2011. Efficient marginal likelihood optimization in blind deconvolution. In *CVPR 2011*, 2657–2664.
- LIU, R., AND JIA, J. 2008. Reducing boundary artifacts in image deconvolution. In *ICIP’08*, 505–508.
- LUCY, L. B. 1974. An iterative technique for the rectification of observed distributions. *The Astronomical Journal* 79, 745+.
- LUKAS, M. A. 2006. Robust generalized cross-validation for choosing the regularization parameter. *Inverse Problems* 22, 5, 1883.
- SHAN, Q., XIONG, W., AND JIA, J. 2007. Rotational motion deblurring of a rigid object from a single image. In *ICCV 2007*, 1–8.
- SHAN, Q., JIA, J., AND AGARWALA, A. 2008. High-quality motion deblurring from a single image. *ACM TOG* 27.
- SIBARITA, J. B. 2005. Deconvolution microscopy. *Adv Biochem Eng Biotechnol.*, 95, 201–243.
- STARCK, J. L., PANTIN, E., AND MURTAGH, F. 2002. Deconvolution in astronomy: A review. *Publications of the Astronomical Society of the Pacific* (October), 1051–1069.
- STEWART, C. V. 1999. Robust parameter estimation in computer vision. *SIAM Reviews* 41, 513–537.

Image #	Blurry	Lucy	Zhou	ℓ_2	IRLS $\alpha = 4/5$	TV	ℓ_1	Shan	Krishnan <i>LUT</i> $\alpha = 2/3$	Our
01	21.24	23.71	26.46	26.09	26.81	26.92	26.11	26.84	27.00	27.23
02	27.83	24.64	28.42	28.07	32.15	30.49	29.39	31.92	32.01	32.25
03	28.18	24.65	28.60	28.22	33.07	30.98	29.80	32.81	32.88	33.18
04	27.00	24.59	28.51	28.12	32.05	30.34	29.27	31.72	31.78	32.04
05	20.05	23.91	26.22	26.09	26.91	27.30	26.36	26.89	27.17	27.34
06	23.23	24.29	27.14	26.81	27.85	27.79	26.98	27.90	28.01	28.18
07	24.63	24.46	28.44	28.05	31.66	30.22	29.10	31.02	31.06	32.08
08	18.30	23.71	25.49	25.27	26.30	26.41	25.56	26.33	26.71	26.58
09	25.49	24.53	28.39	27.97	32.41	30.49	29.32	32.12	32.19	32.57
10	26.19	24.60	28.47	28.06	32.57	30.65	29.47	32.23	32.36	32.44
11	23.85	24.55	27.57	27.27	29.49	28.96	28.00	29.48	29.66	29.73
12	27.60	24.65	28.47	28.10	32.56	30.71	29.61	32.30	32.42	32.73
13	19.64	23.69	25.17	25.11	25.04	25.85	24.98	25.34	25.62	25.20
14	22.94	24.38	27.50	27.22	29.02	28.60	27.69	28.93	29.04	29.16
15	27.06	25.09	28.63	28.34	32.53	30.77	29.68	32.25	32.34	32.53
16	26.74	24.35	28.09	27.69	30.38	29.43	28.33	30.30	30.34	30.60
17	25.91	24.81	28.31	27.97	31.56	30.25	29.13	31.39	31.56	31.66
18	23.06	24.43	27.29	27.03	28.70	28.52	27.54	28.77	28.92	28.74
19	22.42	24.29	27.36	26.96	29.13	28.54	27.59	28.99	29.30	29.63
20	25.41	26.15	29.31	29.02	32.42	31.41	30.44	32.24	32.47	32.54
21	23.04	24.38	27.38	27.04	28.96	28.63	27.68	28.97	29.15	29.25
22	24.88	24.58	27.93	27.63	30.33	29.42	28.45	30.24	30.33	30.36
23	27.25	24.67	28.68	28.30	33.80	31.26	30.00	33.27	33.54	34.13
24	22.00	24.33	26.69	26.51	27.98	28.21	27.13	28.17	28.52	28.09
Av. PSNR gain		0.15	3.36	3.04	5.82	4.92	3.90	5.69	5.89	6.01
Av. Time (secs)		5.45	1.87	0.50	626.56	5.51	62.19	8.22 (exe*)	1.99 (mex*) 3.33	0.82

Table 1 PSNR and running time for nine non-blind deconvolution techniques applied to the 24 images from the Kodak Lossless True Color Image Suite [15]. Higher PSNR highlighted in bold. The input images were blurred using the 19×19 blur kernel of Krishnan and Fergus [16] and received 1% Gaussian white noise. The image numbers match the numbers in the file names (kodim#.png). * exe and mex: Pre-compiled C code (all other measurements and techniques use pure MATLAB code).

Kernel #/ size	Blurry	Lucy	Zhou	ℓ_2	IRLS $\alpha = 4/5$	TV	ℓ_1	Shan	Krishnan <i>LUT</i> $\alpha = 2/3$	Our
#1: 13×13	29.54	24.44	28.95	28.43	33.61	31.29	30.01	30.09	33.37	33.73
#2: 15×15	29.01	24.64	28.88	28.46	33.40	31.43	30.41	30.24	33.25	33.46
#3: 17×17	28.57	24.81	28.80	28.46	33.17	31.41	30.48	30.05	33.04	33.26
#4: 19×19	28.18	24.65	28.60	28.23	33.06	30.98	29.79	32.81	32.87	33.18
#5: 21×21	27.85	25.08	28.52	28.39	32.70	31.48	30.80	32.62	32.63	32.80
#6: 23×23	27.53	25.49	28.41	28.43	32.48	31.64	31.05	32.40	32.41	32.56
#7: 27×27	26.96	26.40	28.36	28.56	32.11	31.73	31.23	32.04	32.00	32.20
#8: 41×41	25.39	28.24	28.08	28.70	31.13	31.43	30.43	29.69	30.89	31.24
Av. PSNR gain		-2.41	0.70	0.58	4.83	3.55	2.65	3.36	4.68	4.93
Av. Time (secs)		5.32	1.73	0.47	451.95	5.91	58.21	8.05	1.96	0.80

Table 3 PSNR and running times for nine non-blind deconvolution techniques applied to the original (sharp) version of the image shown in Fig. 2, blurred with eight different kernel sizes plus 1% Gaussian noise. Our method achieves higher PSNR for seven of the eight kernel sizes. It is also faster than all methods, except for ℓ_2 , but ours achieves significantly higher PSNR.

30. YU-WING TAI; XIAOGANG CHEN; SUNYEONG KIM; SEON JOO KIM; FENG LI; JIE YANG; JINGYI YU; MATSUSHITA, Y.; BROWN, M.S., 2013. Nonlinear Camera Response Functions and Image Deblurring: Theoretical Analysis and Practice. IEEE TPAMI. 2013, 35, 2498-2512.
31. TIKHONOV, A. N., AND ARSENIN, V. Y. 1977. *Solutions of Ill-Posed Problems*. ohn Wiley & Sons, New York.
32. TOMASI, C., AND MANDUCHI, R. 1998. Bilateral filtering for gray and color images. *ICCV '98*, 839-846.
33. WANG, Y., YANG, J., YIN, W., AND ZHANG, Y. 2008. A new alternating minimization algorithm for total variation

- image reconstruction. *SIAM J. Img. Sci.* 1, 3, 248-272.
34. YUAN, L., SUN, J., QUAN, L., AND SHUM, H.-Y. 2008. Progressive inter-scale and intra-scale non-blind image deconvolution. *ACM Trans. Graph.* 27, 3, 74:1-74:10.
35. XU, L. AND JIA, J. 2010. Two-phase Kernel Estimation for Robust Motion Deblurring *ECCV'10*, 157-170.
36. XU, L. AND ZHENG S. AND JIA, J. 2013. Unnatural L0 Sparse Representation for Natural Image Deblurring *CVPR, 2013* 1107-1114.
37. ZHOU, C., AND NAYAR, S. K. 2009. What are good apertures for defocus deblurring? In *ICCP*, 1-8.