

# BLIND DEBLURRING OF NATURAL IMAGES

Mariana S. C. Almeida and Luís B. Almeida

Instituto de Telecomunicações and Instituto Superior Técnico, Lisbon, Portugal  
(mariana.almeida, lbalmeida)@lx.it.pt

## ABSTRACT

A new method to perform blind image deblurring is proposed. Very few assumptions are made on the blurring filter and on the original image: the blurring filter is assumed to have limited support and the original image is assumed to be a sharp natural image. A new prior is used, which gives higher probability to images with sharp edges. The estimation of both the deblurred image and the blurring filter is made in a progressive way, first taking into account the main features of the image, and then proceeding to smaller details.

The results obtained with synthetically blurred images are good, even when the blur operator is rather ill-conditioned and the blurred image is noisy. The method also yields improvements in real-life photographs with focus and motion blurs.

**Index Terms**— Blind image deconvolution, Image enhancement, Image restoration, Image edge analysis, Sparse distributions

## 1. INTRODUCTION

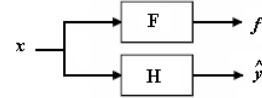
Image deblurring consists of attempting to recover an image which has been degraded by a linear shift-invariant filtering operation, possibly with noise. This has applications in fields such as astronomy [1], remote sensing [2] and biomedical imaging [3], as well as in everyday life, for the enhancement of blurred photos.

Part of the previous work on image deconvolution has been done in the non-blind setting, in which the blur operator is exactly known. Although that is not a valid assumption in many real-life situations, the problem is still hard, because the blur operator typically is very ill-conditioned. Several approaches to this problem, using prior information on the estimated image, can be found in [4, 5, 6].

In blind image deconvolution (BID), both the image and the blur operator are unknown. The problem is ill-posed, having an infinite number of solutions. Furthermore, as above, the blur operator often is rather ill-conditioned. For an overview of BID see [7].

To the authors' knowledge, there is no previous general solution for blind deblurring without making relatively strong assumptions on the blurring operator and/or on the image. A common approach is to restrict the problem to some blur model, such as motion blur or focus blur [8, 9]. Among recent works on BID we emphasize [10, 11, 12, 13]. Both [10] and [11] require extra data for a preliminary training. [12] attempts to encompass less restrictive blurs through a fuzzy technique, which is applied under the output of blur models known a priori. A non-iterative, fast method with proof of convergence is presented in [13]. This method assumes the blur to be zero-phase and depends on a good initial estimate of the blur.

The approach that we propose in this paper makes very few assumptions on the blurring filter: only that it has a support of size no



**Fig. 1.** Schematic representation of the deblurring method. Block  $F$  extracts edge intensities.

larger than a given value and that it generally has a low-pass character. We use maximum a posteriori approach with a new prior which favors images with sharp edges. This leads to a regularizer which generalizes the well known total variation (in its discrete form) [14].

Directly estimating the image using the new prior doesn't easily lead to good solutions. We adopt a learning technique which guides the optimization to a good solution by first concentrating on the main features of the image, and progressively dealing with smaller details. This leads to quite good results, even for blur operators that are rather ill-conditioned.

This paper is structured as follows: Section 2 describes the proposed method and introduces the new regularizer. Results are presented in Section 3. Section 4 concludes.

## 2. DEBLURRING METHOD

The image degradation can be formulated as

$$y = Hx + n, \quad (1)$$

where  $H$  is a square matrix corresponding to the linear blurring operator, and  $y$ ,  $x$  and  $n$  are column vectors that represent, respectively, the degraded image, the original image and additive noise, all vectorized in lexicographic order.

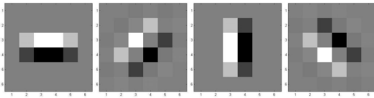
The method that we propose can be explained with the aid of Fig. 1, where  $x$  is the estimated deblurred image,  $H$  the estimated blur filter and  $\hat{y}$  a reconstructed approximation to the blurred image. Block  $F$  represents an edge extractor: for each pixel of the image  $x$ , it computes the intensity of a possible edge passing through that pixel. The estimation of both  $x$  and  $H$  is performed through the minimization of the objective function

$$C(x, H) = \|y - Hx\|_2^2 + \lambda R(F(x)), \quad (2)$$

where  $\lambda$  in (2) is a regularization parameter and  $R(F(x))$  is a regularizing term which favors solutions in which the deblurred image  $x$  has a sparse response to the edge extractor  $F$ . More details on  $F$ , on the prior and on the regularizing term  $R(F(x))$  are given in Sections 2.1 and 2.2.

The blurring filter  $H$  is restricted to a limited support. The learning is guided to the desired solution by starting with a large regularization parameter  $\lambda$ , which is then progressively reduced, and by

This work was partially supported by the FCT project POSC/EEACPS/61271/2004 and by FCT's PhD fellowship SFRH/BD/23919/2005.



**Fig. 2.** The set of edge detection filters, in the four orientations that were used.

starting with a less sparse prior, which is then made sparser. For efficiency reasons, optimizations relative to  $x$  and  $H$  are performed in alternation. An outline of the method is shown in Table 1. We assume that we have chosen a decreasing sequence of values for  $\lambda$  and a sequence of values for the prior’s sparsity which increases or stays constant from one step to the next.

*Initialization:*

- 1 – Set  $H$  to the identity operator.
- 2 – Set  $x$  equal to  $y$ .
- 3 – Set  $\lambda$  and the prior’s sparsity to the initial values of the corresponding sequences.

*Optimization loop:*

- 4 – Find new  $x$  estimate:  $x = \operatorname{argmin}_x C(x, H)$  ( $H$  fixed).
- 5 – Find new  $H$  estimate:  $H = \operatorname{argmin}_H C(x, H)$  ( $x$  fixed).
- 6 – Set  $\lambda$  and the prior’s sparsity to the next values in sequence.
- 7 – If  $\lambda \geq \lambda_{\min}$  go back to 4; otherwise stop.

**Table 1.** Deblurring method

In the beginning of the optimization, with a large  $\lambda$ , only the main features survive in the estimated image. It makes sense to start the optimization by considering only these features. In fact, while the estimate of  $H$  is poor, an image estimate with little regularization would contain many wrong high frequencies, and only the stronger and largest features would survive. By using a strong regularization we force the process to only consider those features, eliminating the wrong high frequencies. As the optimization proceeds and the estimate of the blurring filter becomes better, smaller and fainter features can progressively be used for the estimation. By progressively lowering  $\lambda$  during the optimization, we guide the method to progressively consider smaller features, leading it to a good solution. An early image estimate, with a large  $\lambda$ , is shown in Fig. 3 b).

A very sparse prior doesn’t easily allow edges to move. If, initially, the estimated position of an edge is slightly offset relative to the correct position, it will be hard for the edge to move to the right position during the optimization. For this reason, we start the optimization process with a prior that is less sparse than the one used in the final phases of the optimization.

### 2.1. Edge detector

The edge detector uses a set of edge detection filters, which are rotated versions of a basic filter (see Fig. 2). Filter rotation is performed by rotating the basic filter’s point spread function, using bicubic interpolation. For each pixel, the edge detector computes the outputs of the filters for all orientations under consideration,  $g_\theta$ , where  $\theta$  denotes the filter’s orientation. The detector’s output is given by

$$f = \sqrt{\sum_{\theta \in \Theta} g_\theta^2}, \quad (3)$$

where  $\Theta$  is the set of orientations under consideration.

### 2.2. Image prior

Edges, in natural images, are known to be sparse. They are sparser in sharp images than in blurred ones, because, in the latter, edges

are “spread” over a larger width. Therefore, a sparse prior on edges will favor sharp images over blurred ones. The prior that we use assumes 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 is assumed to follow a sparse prior with density

$$p(f) \propto e^{-k(f+\epsilon)^q}, \quad (4)$$

where  $k$  adjusts for the scale of edge intensities and  $q$  controls the prior’s sparsity;  $\epsilon$  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

$$p(x, H|y) \propto e^{-\frac{1}{2\sigma^2} \|Hx - y\|_2^2} \prod_i e^{-k(f_i + \epsilon)^q}, \quad (5)$$

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

$$L(x, H|y) = -\frac{1}{2\sigma^2} \|Hx - y\|_2^2 - k \sum_i (f_i + \epsilon)^q. \quad (6)$$

Maximizing this likelihood is equivalent to minimizing the cost function

$$C(x, H) = \|Hx - y\|_2^2 + \lambda \sum_i (f_i + \epsilon)^q, \quad (7)$$

where  $\lambda = k2\sigma^2$ . This cost function is of the form given in (2).

We recognize, in this equation, the data connection term,  $\|Hx - y\|_2^2$ , and the regularizer,  $\sum_i (f_i + \epsilon)^q$ . The latter has the well known total variation regularizer (in its discrete form) as a special case, which is obtained by using just two filters which compute horizontal and vertical differences, and setting  $\epsilon = 0$  and  $q = 1$ .

As was said above, we decrease  $\lambda$  during the optimization. Therefore, except for the last phase of the optimization,  $\lambda$  is not given by  $k2\sigma^2$ . And in fact, even during that last phase,  $\lambda$  still is not given by that expression because the noise  $n$ , besides allowing for possible noise in the blurred image, actually 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 actual one, this difference being treated by the method as noise.

## 3. EXPERIMENTS

In this section we present several practical examples illustrating the performance of the proposed method. The first four experiments involve synthetic blurs and the last two involve actual blurred photos.

We have used, in the experiments, the method described above, outlined in Table 1. The support of the blurring filter estimate was limited to a square of size  $s \times s$  pixels (the specific values of  $s$  are given ahead for each case). In step 5 of the method, where we estimate the blurring filter, we used a safety margin of width 3 around that square. This means that the filter’s support was limited to a size  $(s + 6) \times (s + 6)$  and that, at the end of step 5, the filter estimate was truncated to the central square of size  $s \times s$ . This safety margin was used because we found that there were some undesirable border effects in the filter estimate, in the initial phases of the optimization.

The sequences of values of  $\lambda$  and  $q$  were set, for all experiments, at 2, 0.6, 0.2, 0.06,  $\dots$ ,  $2 \times 10^{-7}$  and 0.8, 0.8, 0.6, 0.6, 0.6, 0.4, 0.4,  $\dots$ , 0.4, respectively (a total of 15 values each). In

the tests that involved noise (artificial or natural) these sequences were truncated, so that the final value of  $\lambda$  was larger. Parameter  $\epsilon$  was set to 0.002. Pixel intensities were represented on a scale of 0 (black) to 1 (white). All images had a size of  $256 \times 256$  pixels. The edge detection filters that we used are shown in Fig. 2. They correspond to the basic filter with point spread function (PSF)

$$F_{\theta=0} = \begin{bmatrix} 1 & 2 & 2 & 1 \\ -1 & -2 & -2 & -1 \end{bmatrix} / 12, \text{ with rotations of } 45^\circ.$$

The results of the first four experiments are shown in Figs. 3 and 4. In the printed version of the paper the images are small, but in the electronic version one can zoom in on the images to see finer details. In the first experiment the blur PSF was a uniform square of size  $9 \times 9$ . In the second one it corresponded to a simulated motion blur of length 9 pixels, in a direction at  $45^\circ$  with the horizontal. In the third one it was a simulated defocus blur (a uniform circle of diameter 11 pixels). The size of the support of the estimated filter ( $s$ , above) was 15, 15 and 17, respectively. The fourth experiment was similar to the third, but Gaussian noise with  $\sigma = 0.01$  (SNR of  $25dB$ ) was added to the blurred image. In this case, the sequence of values of  $\lambda$  was truncated at 0.0006, followed by a last iteration with  $\lambda = 0.0004$ . This last value was found, by visual inspection, to provide the best trade-off between sharpness and noise in the deblurred image.

The last two experiments, whose results are shown in Fig. 5, involved actual photographs, intentionally taken with a defocus and a motion blur, respectively. The sequence of values of  $\lambda$  was truncated at 0.0002. In the case of the motion blur, a further iteration with  $\lambda = 0.0001$  was performed. Again, these values were experimentally found. The filter size  $s$  was set to 15 in both cases.

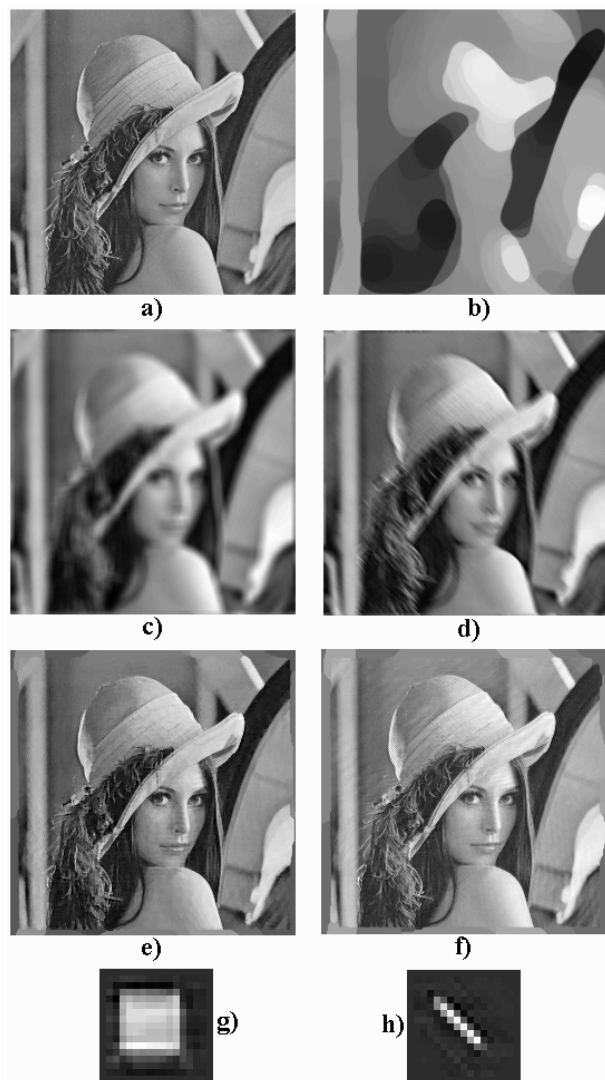
These results show that, in synthetic blurs with no noise, the estimates of the original image and of the blurring filter were rather good. Even in the noisy case, the recovered image was significantly sharper than the blurred one and didn't have too much noise, although the filter estimate was rather noisy. In the case of the photographs, the recovered images had somewhat a synthetic look and presented some artifacts, but they were significantly sharper and had more details than the blurred ones. The somewhat lower quality of the results with photos probably means that the blurs didn't exactly follow the model (1), perhaps due to nonlinearities in the camera's processing and to the JPEG encoding of the photos.

The optimization of the deblurred image  $x$  was performed by gradient descent with adaptive step sizes (150 iterations for each value of  $\lambda$ ). The filter  $H$  was optimized by conjugate gradients (100 iterations for each value of  $\lambda$ ). On an Intel Core 2 Duo system running at 2 GHz, programmed in Matlab and running only on one of the chip's processors, the optimization took about 30 seconds for each value of  $\lambda$  (7.5 minutes for the 15 values).

#### 4. CONCLUSIONS AND FUTURE WORK

A new method for blind image deblurring was proposed. The only strict assumption of the method is that the blurring filter has limited support, no larger than a given size. A new prior is used, which tends to enforce the sharpness of edges, by making them sparse. Edges are found by means of a new detector, which uses several rotated versions of a basic edge-detection filter. The learning is guided to a reasonable solution by first concentrating on the main features of the image, and progressively taking into account smaller details. Experimental results, both on synthetic blurs and on actual blurred photographs, show the good performance of the method.

The prior that was introduced in this paper will probably be useful in other image restoration problems as well. Quite probably, it will also be possible to improve the prior itself. We also plan to ex-



**Fig. 3.** Results of synthetic experiments. a) Original image. b) Estimate for the first  $\lambda$  value, for the square blur. Next rows: left, square blur; right, motion blur. Second row: Blurred images. Third row: Deblurred images. Fourth row: Filter estimates.

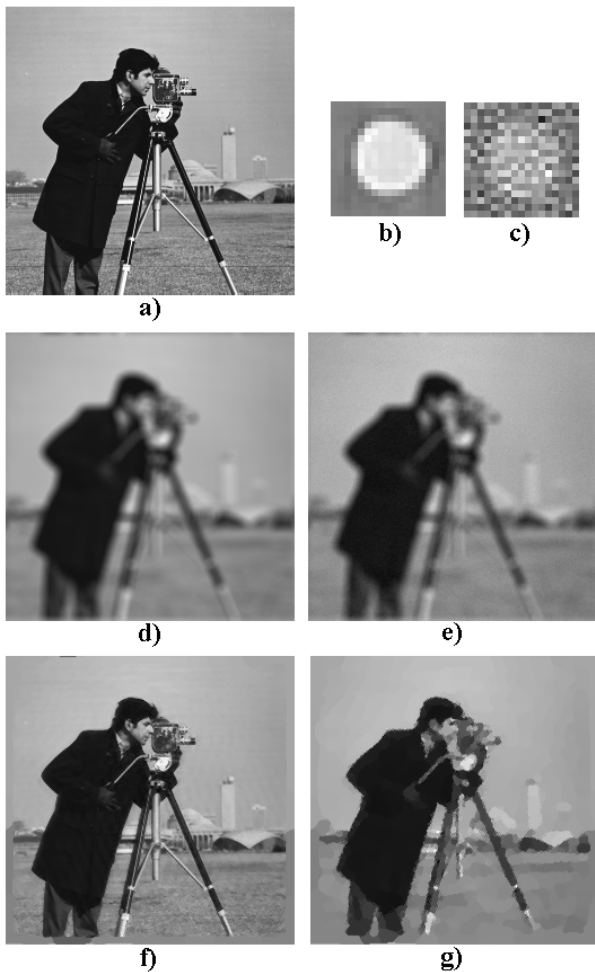
plore efficient techniques, such as majoration-minimization, to improve the optimization speed of the method.

#### 5. ACKNOWLEDGMENTS

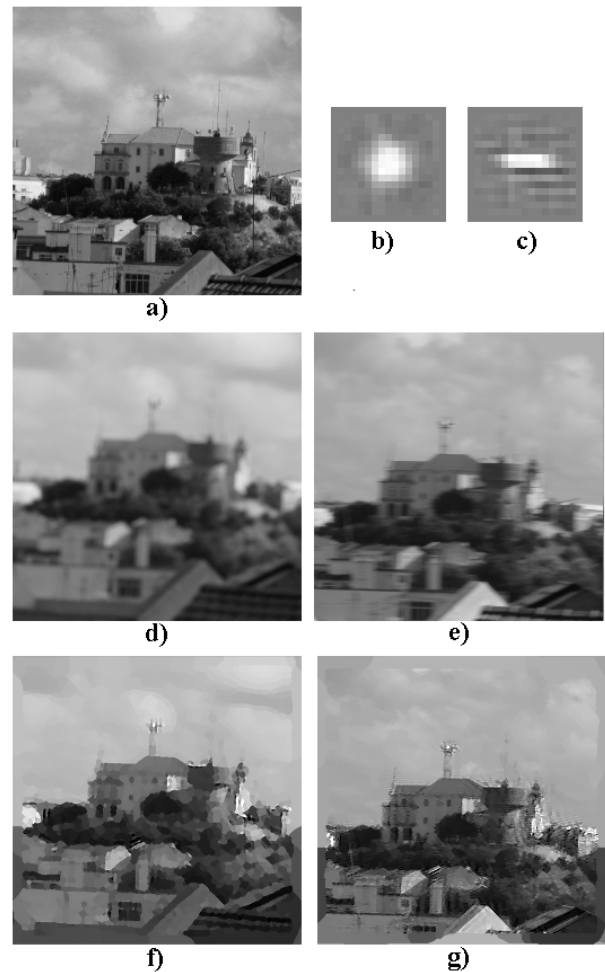
We wish to acknowledge very useful discussions with our colleagues J. Bioucas-Dias and M. Figueiredo.

#### 6. REFERENCES

- [1] M. Bertero and P. Boccacci, "Image restoration methods for the large binocular telescope," *Astron. Astrophys. Suppl.*, vol. 147, pp. 323–333, 2000.
- [2] J. P. Muller, *Digital Image Processing in Remote Sensing*, Taylor & Francis, 1988.



**Fig. 4.** Results of synthetic experiments. a) Original image. b) and c) Filter estimates. Next rows: Left, without noise; Right, with noise. Second row: Blurred images. Third row: Deblurred images.



**Fig. 5.** Results with actual blurred photos. a) Sharp photo. b) and c) Filter estimates. Next rows: Left, defocus blur; Right, motion blur. Second row: Blurred images. Third row: Deblurred images.

- [3] D. Adam and O. Michailovich, "Blind deconvolution of ultrasound sequences using non-parametric local polynomial estimates of the pulse," *IEEE Trans. on Biomedical Eng.*, vol. 42, no. 2, pp. 118–131, 2002.
- [4] G. K. Chantas, N. P. Galatsanos, and A. C. Likas, "Bayesian restoration using a new nonstationary edge-preserving image," *IEEE Trans. on Image Processing*, vol. 15, no. 10, 2006.
- [5] J. A. Guerrero-Colon and J. Portilla, "Deblurring-by-denoising using spatially adaptive gaussian scale mixtures in overcomplete pyramids," in *IEEE International Conference on Image Processing*, Atlanta, October 2006, pp. 625–628.
- [6] J. Bioucas-Dias, M. Figueiredo, and J. Oliveira, "Adaptive bayesian/total-variation image deconvolution: A majorization-minimization approach," in *European Signal Processing Conference*, Florence, Italy, September 2006.
- [7] D. Kundur and D. Hatzinakos, "Blind image deconvolution," *IEEE Sig. Proc. Magazine*, pp. 43–64, May 1996.
- [8] F. Kraher, Y. Lin, B. McAdoo, K. Ott, J. Wang, D. Widmann, and B. Wohlberg, "Blind image deconvolution: motion blur estimation," Tech. Rep., University of Minnesota, 2006.
- [9] J. Oliveira, M. Figueiredo, and J. Bioucas-Dias, "Blind estimation of motion blur parameters for image deconvolution," in *Iberian Conference on Pattern Recognition and Image Analysis*, Girona, Spain, June 2007.
- [10] M. M. Bronstein, M. Bronstein, and M. Zibulevsky, "Blind deconvolution of images using optimal sparse representations," *IEEE Trans. on Image Processing*, vol. 14, pp. 726–735, 2005.
- [11] D. Li, R. M. Mersereau, and S. Simske, "Blind image deconvolution through support vector regression," *IEEE Trans. on Neural Networks*, vol. 18, no. 3, May 2007.
- [12] K.-H. Yap, L. Guan, and W. Liu, "A recursive soft-decision approach to blind image deconvolution," *IEEE Trans. on Signal Processing*, vol. 51, no. 2, February 2003.
- [13] L. Justen and R. Ramlau, "A non-iterative regularization approach to blind deconvolution," *Institute of Physics Publishing Inverse Problems*, vol. 22, pp. 771–800, 2006.
- [14] L. Rudin, S. Osher, and E. Fatemi, "Nonlinear total variation based noise removal algorithms," *Physica D*, vol. 60, pp. 259–268, 1992.