Removing Boundary Artifacts for Real-Time Iterated Shrinkage Deconvolution

Michal Šorel

***Abstract*—We propose a solution to the problem of boundary artifacts appearing in several recently published fast deblurring algorithms based on iterated shrinkage thresholding in a sparse domain and Fourier domain deconvolution. Our approach adapts an idea proposed by Reeves for deconvolution by the Wiener filter. The time of computation less than doubles.**

***Index Terms*—Deblurring, deconvolution, image processing, image restoration, iterated shrinkage thresholding, primal-dual methods, sparsity, Wiener filter.**

I. Introduction

In this paper, we address the classical problem of deconvolution, i.e., to find the original image when we know an observed image blurred by a known blur kernel and degraded by an additive Gaussian noise. We use a matrix notation so that the convolution of image x with kernel h is written as Hx, where H is a block Toeplitz matrix with Toeplitz blocks and x is taken as a column vector got by stacking all columns of the image to one long vector. In this notation, our observation model can be written as $y = Hx + n$, where n is a Gaussian noise of variance σ^2. Matrix H has more columns than rows because observation y includes only pixels not influenced by the unknown area outside of image x.

Deconvolution is usually viewed from the probabilistic viewpoint as a maximum *a posteriori* probability problem, i.e., we look for image x with the highest posterior probability, given an estimate of image prior probability distribution $p(x)$. For Gaussian noise, this is equivalent to minimization

$$\arg\min_x \frac{1}{2\sigma^2}\|y - Hx\|^2 - \log p(x). \tag{1}$$

The prior probability distribution $p(x)$ is never known exactly and must be estimated. In addition, its form must be chosen so that the functional (1) could be minimized efficiently.

If H is a circular convolution and the prior distribution can be expressed as $-\log p(x) = \sum \alpha_j \|l_j * x\|^2$, then (1) can be solved in the Fourier domain exactly as

$$\hat{x} = \frac{\hat{y}\hat{h}^*}{|\hat{h}|^2 + 2\sigma^2 \sum \alpha_j |\hat{l_j}|^2} \tag{2}$$

which is the well-known Wiener filter with signal variance given by the inverse sum of power spectra of kernels l_j. For the Tikhonov regularization with the image gradients, l_1 and l_2 are derivatives in x- and y-directions.

Manuscript received October 27, 2010; revised June 30, 2011 and October 09, 2011; accepted October 24, 2011. Date of publication November 16, 2011; date of current version March 21, 2012. This work was supported by the Czech Ministry of Education, Youth, and Sports under Project 1M0572 (Research Center DAR). The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Farhan A. Baqai.

The author is with the School of Engineering and Physical Sciences, Heriot-Watt University, EH14 4AS Edinburgh, U.K. (e-mail: sorel@utia.cas.cz).

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.2011.2176344

The main problem of the implementation in the Fourier domain is the introduction of boundary artifacts caused by the fact that H is not circulant. In [1], Reeves described a trick that transforms problem (1) with such noncirculant H to the circular deconvolution (2) (for details, see Section IV).

In recent years, a group of techniques has appeared based on the fact that natural images can be decomposed as a linear combination of few atoms from an overcomplete dictionary [2]. The application of sparsity priors to deconvolution provides state-of-the-art results. A significant part of the corresponding research, including recent papers [3]–[8], aims at improving the speed of these algorithms.

Here, we are interested in a so-called analysis-based sparsity approach [9]. The prior probability is described by an L_p norm [1] as $p(x) \propto \exp(-\beta\|\Phi^T x\|_p^p)$, where Φ^T is a linear analysis operator (transform to a sparse domain) such as the gradient, wavelets, or an overcomplete dictionary. Considering the possibility of multiple input images, i.e., distinguished here by index i, this gives the solution of our deconvolution problem as

$$\arg\min_x \beta\|\Phi^T x\|_p^p + \sum \frac{1}{2\sigma_i^2}\|y_i - H_i x\|^2. \tag{3}$$

The standard solutions of (3) with the L_1 norm are based on convex programming and, as such, are relatively time-consuming (for references, see [7]). The development of iterated shrinkage thresholding techniques [8], [10]–[16] provided a convenient tool for simple and fast computation. For a good overview of early synthesis-based techniques (see [17]). Cai *et al.* [16] describes the constrained split Bregman method including convergence proofs for the analysis-based formulation. Faster two-step methods were proposed in [18] and [7], in the latter case including convex constraints and the proved quadratic convergence rate. A method suitable for parallel implementation and hybrid regularizers is given in [19]. For the latest developments in the area of more general primal-dual algorithms, see [8], [20]. Note that for $p < 1$ the functional (3) is not convex; for convergence properties in this case, see [21].

The method presented in this paper relates to a special class of the iterated shrinkage thresholding methods accelerating computation using deconvolution in the Fourier domain [4], [6], [8], [22]. As detailed in Section II, we propose a modification that removes boundary artifacts produced by division in the Fourier domain while keeping the speed close to that of the original methods.

II. Contributions

In the recent methods [4], [16], [22]–[24], the algorithm minimizing (3) alternates between shrinkage thresholding in a sparse domain and deconvolution in the Fourier domain, which converges to a satisfactory solution in a few iterations. The main problem of these algorithms are boundary artifacts, analogous to those we know from Wiener filtering. In this paper, we show how to remove these artifacts without excessive slow down. Results improve significantly in terms of both the visual quality and the mean square error.

Our solution is inspired by an idea published in [1] for the deconvolution with Tikhonov regularization, which transforms the problem (1) with noncirculant H to the circular deconvolution (2), with the blurred

[1]Despite violating the triangle inequality, it is common to refer to $\|x\|_p$ as a norm even for $p < 1$. We use this convention as well.

	Algorithm (boundaries estimated in the first q iterations)
1	Replicate and smooth boundaries of input images y_i
2	$x = y_1$
3	$a = \Theta_{hard}(\Phi^T x, \mu^{-1/2})$ for $p = 0$, or $a = \Theta_{soft}(\Phi^T x, \frac{1}{2}\mu^{-1})$ for $p = 1$
4	if iteration $\leq q$, $u = \left[I - JWBW^T\right]^{-1} JWB(\sum \nu_i H_i^T y_i + \Phi a)$
5	$\hat{x} = \frac{\widehat{\Phi a} + \sum \nu_i \hat{h_i}^* \hat{z_i}}{c + \sum \nu_i \lvert\hat{h_i}\rvert^2}$, where $z_i = P \begin{bmatrix} \nu_i y_i \\ u_i \end{bmatrix}$
6	repeat from 3

Fig. 1. Algorithms I-III for the analysis-based splitting algorithm with tight frame regularization.

input image framed to a previously estimated border of a width corresponding to half of the PSF. The estimation of the border is stated as an inverse problem of significantly smaller dimension.

We extended this idea to the functionals minimized in each iteration of the analysis-based methods [4], [8], [16], [22]. The resulting procedure includes relatively time-consuming estimation of borders that the input images are framed to. If we apply this estimation in each iteration, the artifacts are almost perfectly removed, but the computation is noticeably slower than the original algorithm. In this paper, this algorithm is denoted as Algorithm II (see Fig. 1). To speed-up the algorithm, the estimated borders can be reused during the subsequent iterations of the algorithm, denoted as Algorithm III. In certain cases, it is sufficient to simply treat the boundaries using the MATLAB function edgetaper (part of the Image Processing Toolbox), which blurs the image borders so that the image can be considered approximately periodic (Algorithm I). This is a simpler and faster alternative. However, as demonstrated in the experimental section, the more elaborate solutions (Algorithms II & III) can yield considerably better results, particularly for motion blur and, in general, for images with large differences in the intensity of the opposing sides of the image. The same approach can be used for the augmented Lagrangian method [16], [23] and the primal-dual method [8].

III. Deconvolution by Iterated Shrinkage Thresholding With Analysis-Based Sparsity Priors

In this section, we explain the idea of the analysis-based algorithms [4], [22], mentioned above. Convergence is analyzed in [4], [16], and [21]. All results are shown for an arbitrary number of input images, denoted as k in the sequel.

Our task is to minimize the functional (3). For this, we split variable x and estimate

$$\min_{a,x} \beta \|a\|_p^p + \lambda \|\Phi^T x - a\|^2 + \sum \frac{1}{2\sigma_i^2} \|y_i - H_i x\|^2 \tag{4}$$

where a represents the unknown image x in the sparse domain Φ. The middle term binds x and a together. Equation (4) is equivalent to (3) for λ going to infinity. This equation is alternately minimized with respect to x and its sparse representation a.

A drawback of this approach is that to obtain the solution of the original problem (3), parameter λ must go to infinity, which makes the intermediate minimization increasingly ill-conditioned, thus causing numerical problems [25]. One solution to this problem is to increase λ with each iteration (continuation approach), another is to extend the functional to get to the right solution even with a constant weight on the binding term, which is the case of the split Bregman method [23] or the equivalent augmented Lagrangian method [25].

The simplest solution, i.e., adopted in [4] and [22] and also in our experiments, is to set a reasonably large constant λ and to solve (4) instead of the original problem. In practice, its results are comparable or even better than in the original formulation, and its convergence is sufficient even for $p < 1$. The reason is that the "binding" term makes the functional more convex and helps to get closer to the real minimum.

First, let us treat the minimization of (4) with respect to a (two left terms). This is a simple equation that can be solved analytically and separately in each pixel. For $p = 0$, this is equivalent to the hard thresholding [17] $a = \Theta_{\mathrm{hard}}(\Phi^T x, \mu^{-1/2})$, where $\Theta_{\mathrm{hard}}(\omega, \theta) = 0$ for $|\omega| < \theta$ and $\Theta_{\mathrm{hard}}(\omega, \theta) = \omega$, if otherwise. The second parameter is threshold $\mu = \lambda/\beta$. For $p = 1$, we get the soft thresholding $a = \Theta_{\mathrm{soft}}(\Phi^T x, (1/2)\mu^{-1})$ defined by the continuous function $\Theta_{\mathrm{soft}}(\omega, \theta) = \mathrm{sgn}(\omega)\max(0, |\omega| - \theta)$. A similar thresholding function can be constructed for an arbitrary other L_p norm by a suitable approximation [17].

For now, let us assume that $\Phi\Phi^T$ is diagonal in the Fourier domain. This assumption clearly holds for tight frames, where $\Phi\Phi^T = cI$, with $c > 0$. The second important case is for $\|\Phi^T x\|_p^p$ defined as a concatenation of convolutions with sparsifying kernels $\sum \alpha_j \|l_j * x\|_p^p$. A typical example is total-variation (TV) regularization in its anisotropic form, [2] where l_j are derivatives in the x- and y-directions and $\Phi\Phi^T$ is the convolution with the discrete Laplace operator. However, there are many other possibilities how to choose the sparsifying kernels, such as a suitable subset of wavelets. To have $\Phi\Phi^T$ diagonalizable, we need the convolutions with l_j to be defined as circular. Luckily, as the support of these kernels is small, and the borders are finally removed, taking the convolution circular does not introduce visible artifacts.

If also H_i were simple circular convolutions, we could minimize (4) over x exactly in the Fourier domain. For Φ being a tight frame, we get

$$\hat{x} = \frac{\widehat{\Phi a} + \sum \nu_i \hat{h_i}^* \hat{y_i}}{c + \sum \nu_i |\hat{h_i}|^2} \tag{5}$$

where $\nu_i = 1/2\lambda\sigma_i^2$. Likewise, for $\|\Phi^T x\|_p^p$ in the form $\sum \alpha_j \|l_j * x\|_p^p$, we get analogously to (2)

$$\hat{x} = \frac{\widehat{\Phi a} + \sum \nu_i \hat{h_i}^* \hat{y_i}}{\sum \alpha_j |\hat{l_j}|^2 + \sum \nu_i |\hat{h_i}|^2}. \tag{6}$$

Unfortunately, in practice, the convolution is never circular, and such an approach produces artifacts due to boundary effects [see Figs. 2(c) and 4(c)]. In the next section, we show how to compute the minimum of (4) over x using the Fourier transform without introducing the boundary artifacts.

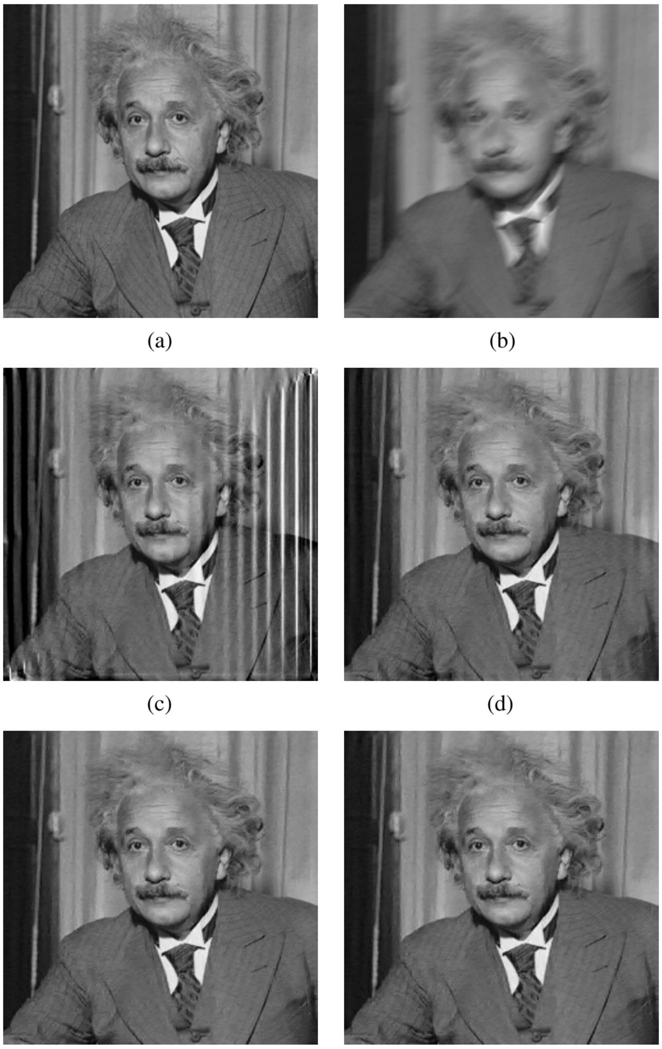
IV. Fourier Domain Deconvolution

In this section, we explain a fast solution to the deconvolution problem, i.e.,

$$\min_x \|\Phi^T x - a\|^2 + \sum \nu_i \|y_i - H_i x\|^2 \tag{7}$$

where $H_i x$ are valid parts of convolutions with PSFs h_i. Therefore, the size of x is larger than y_i, where the difference corresponds to the size of h_i minus one pixel.

[2] Anisotropic regularization computes TVs separately in x- and y-directions. Isotropic formulation takes the integral of the size of discretized image gradient.



(a) (b) (c) (d) (e) (f)

Fig. 2. Comparison of the results of the original algorithm [22], the improved version with boundaries treated by *edgetaper* and more elaborated Algorithms II and III. Artifacts in the form of vertical stripes on the right side of (d) are clearly visible. On the other hand, the difference between Algs. II & III is barely noticeable. **Best viewed electronically**. (a) Original. (b) Blurred (added noise 30 db). (c) Portilla [22]. (d) Algorithm I. (e) Algorithm II. (f) Algorithm III ($q = 1$).

The idea comes from [1], where the author showed how to treat boundary artifacts when deconvolving in the Fourier domain using Tikhonov regularizers, i.e., an approach basically equivalent to the Wiener filter. In [26], the original procedure was applied to deconvolution with regularization using the Huber function. In this section, we modify the original result to work with the left term of (7) and multiple images, which is the form required in algorithms [4], [8], [16], [22]–[24].

Variable a is the known current estimate of the sparse representation of the solution. Therefore, image Φa is also known: It is the synthesis transform of a. The minimizer of (7) is

$$x = \left(\sum \nu_i H_i^T H_i + \Phi\Phi^T\right)^{-1} \left(\sum \nu_i H_i^T y_i + \Phi a\right). \qquad (8)$$

For now, let us assume that $\Phi\Phi^T$ is diagonalizable in the Fourier domain, as described in the end of the previous section. Therefore, if H_i were a circular convolution, we could compute (8) exactly in the Fourier domain by (5) or (6).

Now, we get to the trick. We will find such boundaries u_i of the observed images y_i that would have resulted if image x minimizing (7) had been blurred circularly. Therefore, when circular deblurring (5) is performed on the observation y_i framed to boundaries u_i, the boundary artifacts resulting from the erroneous boundary conditions are eliminated.

To express (8) using circular convolution, we need to define two quantities. First, taking matrix H_i and adding extra rows for the boundary pixels, we can form a new matrix

$$\mathcal{H}_i = P \begin{bmatrix} H_i \\ w_i \end{bmatrix}$$

corresponding to the circular convolution with the same kernel h_i, which is now a block-circulant matrix with circulant blocks. P is a permutation matrix shifting the rows to the appropriate position. Second, let us denote $B = (\sum \nu_i \mathcal{H}_i^T \mathcal{H}_i + \Phi\Phi^T)^{-1}$. It can be shown that the solution of (7) can be written as

$$x = B \left(\sum \mathcal{H}_i^T P \begin{bmatrix} \nu_i y_i \\ u_i \end{bmatrix} + \Phi a\right) \qquad (9)$$

where term $P\left[\begin{smallmatrix} \nu_i y_i \\ u_i \end{smallmatrix}\right]$ corresponds to framing each input image y_i, i.e., multiplied by scalar ν_i, with border u_i. B and $\mathcal{H}_i$ are circulant and therefore (9) can be computed in the Fourier domain by (5) or (6), with the only change that the blurred images y_i are framed to previously computed borders u_i. These borders can be expressed as

$$u = \begin{bmatrix} u_1 \\ \cdot \\ \cdot \\ u_k \end{bmatrix} = [I - JWBW^T]^{-1} JWB \left(\sum \nu_i H_i^T y_i + \Phi a\right) \quad (10)$$

where W is a matrix obtained by stacking all matrices w_i to one matrix

$$W = \begin{bmatrix} w_1 \\ \cdot \\ \cdot \\ w_k \end{bmatrix}$$

and J is a diagonal matrix with ν_i values on the positions corresponding to borders w_i. Multiplication by w_i corresponds to computing circular convolution with h_i taking only boundary pixels. Similarly, $w_i^T z$ corresponds to framing an image of all zeros by boundary pixels z and computing circular convolution by h_i turned $180\,^\circ$ around its center. Obviously, we still need to compute an inversion but of a matrix of much smaller dimension. In our implementation, this inversion is computed by the conjugate gradient method. Due to a low dimension of this problem, we never need more than about three iterations of the conjugate gradients and, as described further, even less when used repetitively in consecutive iterations. The result given in [1] is a special case of (9) and (10) for one image and $a = 0$. The derivation of (9) and (10) is analogous to that in [1].

V. Algorithm

In this section, we detail the proposed algorithm in three versions denoted as Algorithms I–III. All of them are modifications of the iterative procedure described in Section III. At the input, the algorithm needs the blurred images and corresponding PSFs. The user can adjust the number of iterations and parameters λ and β.

The first algorithm (Algorithm I) is basically the original algorithm [4], [22] extended to the possibility of multiple images and with a simple treatment of boundary artifacts. Boundary pixels of the input images are first replicated to achieve the same size as the estimated image x. Then, we apply the MATLAB edgetaper function to smooth

TABLE I
TIME COMPLEXITY AS A NUMBER OF FFTS

Method	FFTs	FFTs for $k = 1$
Algorithm I	$2k + 2n$	$2 + 2n$ (same as [4], [22])
Algorithm II	$2k + (6 + 2k + 4t)n$	$2 + 16n$ ($t = 2$)
Algorithm III	$2k + q(4 + 2k + 4t) + 2n$	$20 + 2n$ ($t = 3$, $q = 1$)

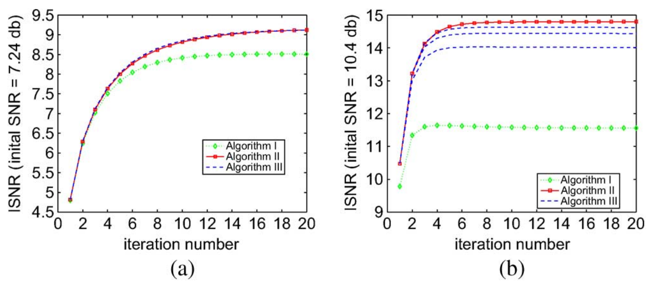


Fig. 3. ISNR for “Einstein” and “Peppers” images in the first 20 iterations. The dashed (blue) line shows results for Algorithm III with boundaries reestimated in the first iteration, which is in the example (a) indistinguishable from the solid (red) line of Algorithm II. In the Peppers graphs, the three dashed (blue) lines denote results for $q = 1 \ldots 3$. The higher q, the better ISNR. (a) Einstein (Fig. 2). (b) Peppers from 2 images (Fig. 4).

the transition between the opposite sides of the images (for a detailed discussion of this function, see [1]). In some cases, this procedure suppresses the artifacts so that the results are acceptable. This is typical for Gaussian-like blurs.

The second version (Algorithm II) treats boundaries rigorously in each main-loop iteration according to formulas derived in Section IV, which results in the almost complete removal of boundary artifacts. Its main disadvantage is a relatively slow speed. The problem is that, although it is usually sufficient to use only two iterations of conjugate gradients to estimate the boundaries (we initialize the boundaries with the value from the previous iteration), it still takes 16 FFTs for a single-image scenario ($k = 1$), which is eight times more than two FFTs in the original algorithms.

Recall that the main motivation is not only to remove the boundary artifacts but also to keep the speed close to that of the original algorithms. Fortunately, the estimation of the border areas need not be very precise to reduce the artifacts. Our experiments showed that it is sufficient to run the estimation of boundaries u_i only in the first one to three iterations (this number is denoted as q) and then reuse them in the subsequent iterations (Algorithm III). This means that except for a small constant number of iterations, our procedure reduces to the computation of almost the same equation as in the original algorithm, which takes just 2 FFTs per iteration. Algorithm III is outlined in Fig. 1. Note that the augmented Lagrangian algorithm has almost the same form, except the addition and subtraction of an auxiliary variable.

Next, we analyze the number of critical time-consuming operations, which is the Fourier transform and transforms to and from a sparse domain. In Table I, we compare the number of FFTs of Algorithm I, which is basically of the same complexity as the original algorithm, with the other two versions. The number of iterations is denoted as n. Algorithm I needs two FFTs per iteration: one to compute $\widehat{\Phi a}$ and the other to return from the Fourier domain after finishing the deconvolution step. In addition, we precompute $\hat{h}_i$ and $\hat{y}_i$ for each input image.

Algorithm II is a complete computation that repeats all steps described in Section IV in each main-loop iteration. In addition to the operations in Algorithm I, Algorithm II requires $4(t + 1)$ FFTs to estimate borders by (10), where t is the number of conjugate gradient iterations. In our experiments, we use $t = 2$, which is sufficient because the borders are being refined during all the consecutive main-loop iterations. The rest of the FFTs are consumed in the precomputations at the beginning of each main-loop iteration.

Fig. 4. Multi-image version of experiment in Fig. 2. Input images [one of them shown in Fig. 4(b)] are blurred by PSFs Fig. 6(b) and (c). Significantly higher ISNR compared to the single-image version can be seen in Fig. 3(b). **Best viewed electronically**. (a) Original. (b) One of two input images (added noise 35 db). (c) Multi-image modification of Portilla [22]. (d) Algorithm I. (e) Algorithm II. (f) Algorithm III ($q = 1$).

Algorithm III computes the boundaries only in the first q iterations, and in the remaining iterations, they are simply reused. In this case, $t = 3$ worked in all examples we tested.

The number of sparsity transforms is $2n$ for all cases (n analysis and n synthesis transforms). Complexity of such transform is usually either $O(N \log N)$ or $O(N)$, where N is the number of pixels. It is important to realize that even the computation of an asymptotically linear transform can take more than an FFT in practice, particularly for highly redundant dictionaries. On the other hand, if we use the gradient as the analysis operator, its computation time is very fast.

For example, for ten iterations, $q = 1$, and for most common frames, the Algorithm III takes only about 50% more time than the original algorithms and less than twice the original time for the regularization by TV. In languages with high overhead of other operations, such as MATLAB, the difference may be even smaller. In our MATLAB implementation, ten iterations of the experiment in Fig. 2 takes 0.17, 0.55

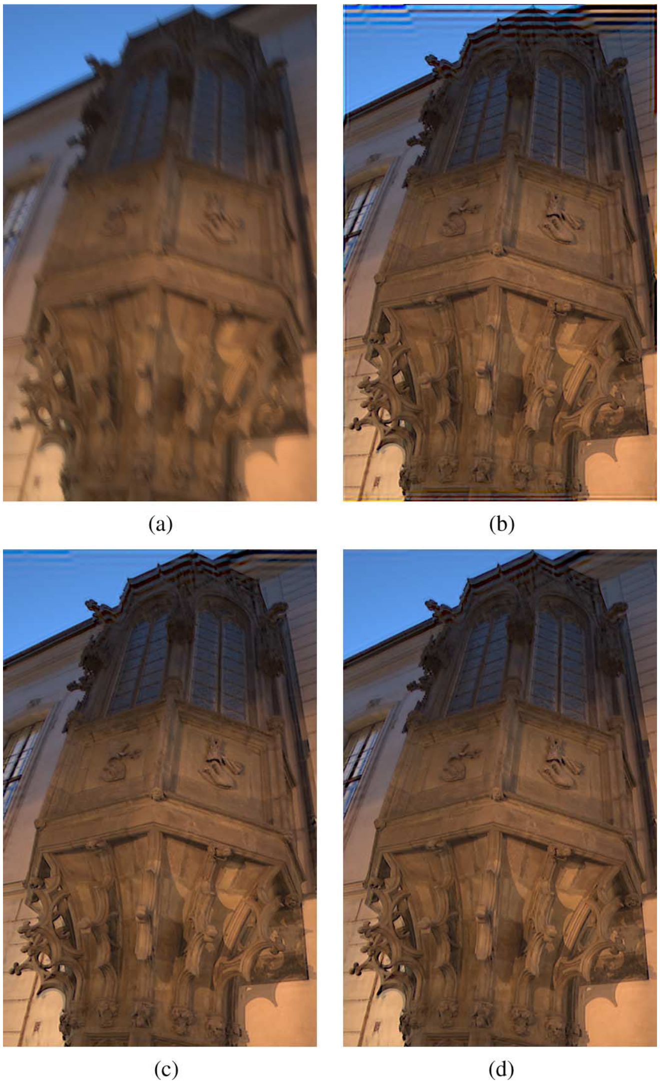


Fig. 5. Night photo (a) was taken from hand with shutter time 2.5 s (1550 $\times$ 980 pixels). The other images show the result of deconvolution (TV regularization, 5 main-loop iterations). **Best viewed electronically**. (a) Blurred image, 3 $\times$ 1.5 MP. (b) Portilla [22]. (c) Algorithm I. (d) Algorithm III ($q = 1$).

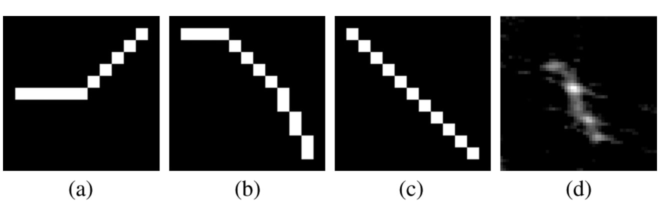


Fig. 6. PSFs used in experiments. PSFs (a), (b) and (c) are artificially generated, PSF (e) of size 51 $\times$ 51 elements was estimated from Fig. 5(a). (a) Einstein. (b) Peppers 1. (c) Peppers 2. (d) Real PSF.

and 0.23 s for Algorithms I, II, and III, respectively (Intel i5, 2.67-GHz processor).

VI. Experiments

For this paper, we chose only the analysis-based splitting approach (as in [4] and [22]) taken with L_0 norm and combined with two types of analysis operators: anisotropic TV and the union of translation-invariant Haar wavelets with dual-tree complex wavelets [27]. The latter

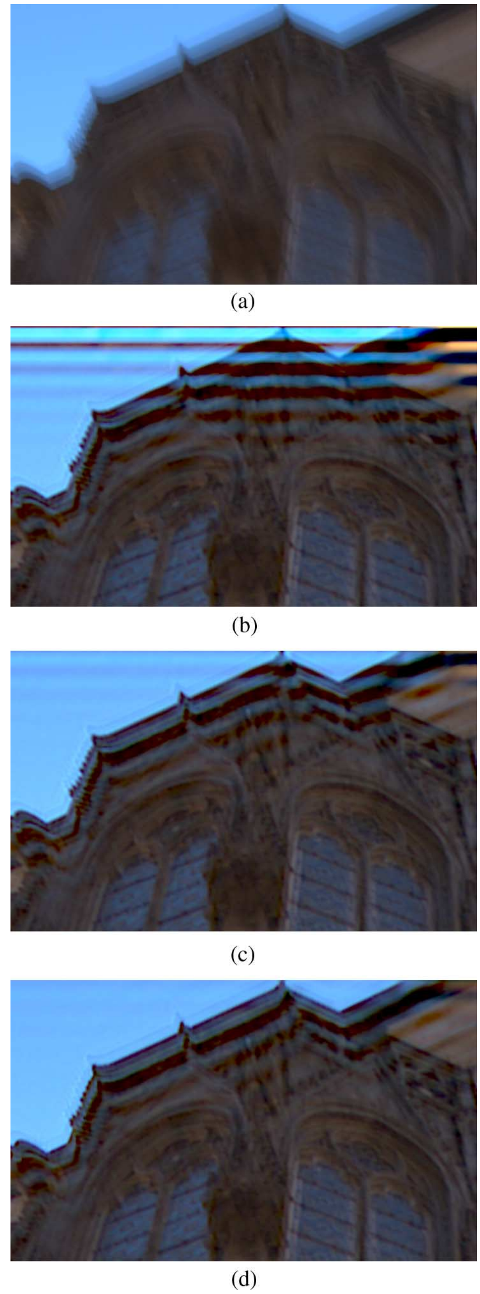


Fig. 7. Close-ups of the top part (300 $\times$ 500 pixels) of Fig. 5. Remaining artifacts are mainly due to spatial variance of the actual PSF. Brightness was adjusted to improve readability in print. **Best viewed electronically**. (a) Blurred image. (b) Portilla [22]. (c) Algorithm I. (d) Algorithm III.

was shown to achieve a state-of-the-art quality of deconvolution in simulated experiments with circular convolution [22]. However, we tested also the L_1 norm and other combinations of sparse transforms with slightly worse results in terms of ISNR but the same conclusions.

Figs. 2 and 3 demonstrate the effect of the better treatment of image boundaries. To evaluate deconvolution error, we work with simulated data. Standard images (Einstein and Peppers) were blurred using artificial kernels [see Fig. 6(a)–(c)] and degraded by additive Gaussian noise (the SNR is 30 and 35 for Figs. 2 and 3, respectively). Fig. 2 uses one input image, and Fig. 3 works with two input images. The figures compare the original Portilla's implementation [22] and the three proposed versions of the boundary treatment. Fig. 4 gives the ISNR as a function of the number of iterations. Notice how the ISNR for Algorithms II and III is almost indistinguishable in Fig. 4.

Fig. 5 is a demonstration of a real use of the proposed approach. To estimate the PSF [see Fig. 6(d)], we took another image with ISO 1600 and 2EV underexposure and, after registration, computed the PSF by the least square fit, as described in [28]. To demonstrate the speed of convergence, we used only five iterations (ten iterations would bring a slight improvement in sharpness) and TV regularization. Note that in all our color experiments, all three color channels were deconvolved separately. (Fig. 7)

VII. Conclusion

This paper presents a technique to remove boundary artifacts in recent nonblind deconvolution algorithms based on iterative shrinkage thresholding in a sparse domain and deconvolution in the Fourier domain. We obtain an excellent quality of restoration, comparable with the current state of the art, while keeping the speed close to the high speed of the original algorithms. Our estimates of achievable frame rates indicate almost real-time performance even for 1-megapixel images on a personal computer.

Although the simple treatment of image boundaries (by edgetaper) can sometimes provide satisfactory results, the proposed solution gives a better alternative particularly for images with large differences in the intensity of the opposing sides of the image or motion PSFs. The proposed algorithm in its full version (Algorithm II) removes boundary artifacts almost perfectly, at the expense of higher time complexity. In addition, in most cases, the much faster approximative algorithm (Algorithm III) removes the artifacts sufficiently even if the boundaries are estimated only in the first main-loop iteration.

The rigorous treatment of boundaries is particularly important when working with small image patches because the importance of boundaries in such cases increases. One example is deblurring in the presence of space-variant blur solved by dividing the image into smaller blocks.

Acknowledgment

The author would like to thank J. Portilla for providing the code of his method.

References

[1] S. J. Reeves, "Fast image restoration without boundary artifacts," *IEEE Trans. Image Process.*, vol. 14, no. 10, pp. 1448–1453, Oct. 2005.

[2] J.-L. Starck, M. Elad, and D. Donoho, "Redundant multiscale transforms and their application for morphological component separation," *Adv. Imaging Electron Phys.*, vol. 132, pp. 287–348, 2004.

[3] J. M. Bioucas-Dias and M. Figueiredo, "An iterative algorithm for linear inverse problems with compound regularizers," in *Proc. 15th IEEE ICIP*, Oct. 2008, pp. 685–688.

[4] Y. Wang, J. Yang, W. Yin, and Y. Zhang, "A new alternating minimization algorithm for total variation image reconstruction," *SIAM J. Imaging Sci.*, vol. 1, no. 3, pp. 248–272, Jul. 2008.

[5] Y.-W. Wen, M. K. Ng, and W.-K. Ching, "Iterative algorithms based on decoupling of deblurring and denoising for image restoration," *SIAM J. Sci. Comput.*, vol. 30, no. 5, pp. 2655–2674, Jun. 2008.

[6] M. Figueiredo, J. Bioucas-Dias, and M. Afonso, "Fast frame-based image deconvolution using variable splitting and constrained optimization," in *Proc. IEEE/SP 15th Workshop SSP*, Aug. 2009, pp. 109–112.

[7] A. Beck and M. Teboulle, "Fast gradient-based algorithms for constrained total variation image denoising and deblurring problems," *IEEE Trans. Image Process.*, vol. 18, no. 11, pp. 2419–2434, Nov. 2009.

[8] A. Chambolle and T. Pock, "A first-order primal-dual algorithm for convex problems with applications to imaging," *J. Math. Imaging Vis.*, vol. 40, no. 1, pp. 120–145, May 2011.

[9] M. Elad, P. Milanfar, and R. Rubinstein, "Analysis versus synthesis in signal priors," *Inverse Probl.*, vol. 23, no. 3, pp. 947–968, Jun. 2007.

[10] G. Chen and M. Teboulle, "A proximal-based decomposition method for convex minimization problems," *Math. Program.*, vol. 64, no. 1, pp. 81–101, Mar. 1994, 10.1007/BF01582566.

[11] R. Nowak and M. Figueiredo, "Fast wavelet-based image deconvolution using the EM algorithm," in *Conf. Rec. 35th Asilomar Conf. Signals, Syst., Comput.*, 2001, vol. 1, pp. 371–375.

[12] J.-L. Starck, M. K. Nguyen, and F. Murtagh, "Wavelets and curvelets for image deconvolution: A combined approach," *Signal Process.*, vol. 83, no. 10, pp. 2279–2283, Oct. 2003.

[13] I. Daubechies, M. Defrise, and C. De Mol, "An iterative thresholding algorithm for linear inverse problems with a sparsity constraint," *Commun. Pure Appl. Math.*, vol. 57, no. 11, pp. 1413–1457, Nov. 2004.

[14] P. L. Combettes and V. R. Wajs, "Signal recovery by proximal forward-backward splitting," *Multiscale Model. Simul.*, vol. 4, no. 4, pp. 1168–1200, 2005.

[15] M. Elad, "Why simple shrinkage is still relevant for redundant representations?," *IEEE Trans. Inf. Theory*, vol. 52, no. 12, pp. 5559–5569, Dec. 2006.

[16] J.-F. Cai, S. Osher, and Z. Shen, "Split Bregman methods and frame based image restoration," *Multiscale Model. Simul.*, vol. 8, no. 2, pp. 337–369, 2009.

[17] M. Elad, B. Matalon, J. Shtok, and M. Zibulevsky, "A wide-angle view at iterated shrinkage algorithms," in *Proc. SPIE Conf. Ser.*, D. V. D. Ville, V. K. Goyal, and M. Papadakis, Eds., Oct. 2007, vol. 6701, no. 1, p. 670 102.

[18] J. Bioucas-Dias and M. Figueiredo, "A new twist: Two-step iterative shrinkage/thresholding algorithms for image restoration," *IEEE Trans. Image Process.*, vol. 16, no. 12, pp. 2992–3004, Dec. 2007.

[19] N. Pustelnik, C. Chaux, and J. Pesquet, "Parallel proximal algorithm for image restoration using hybrid regularization," *IEEE Trans. Image Process.*, vol. 20, no. 9, pp. 2450–2462, Sep. 2011.

[20] E. Esser, X. Zhang, and T. F. Chan, "A general framework for a class of first order primal-dual algorithms for convex optimization in imaging science," *SIAM J. Imaging Sci.*, vol. 3, no. 4, pp. 1015–1046, 2010.

[21] H. Attouch, "Convergence of descent methods for semi-algebraic and tame problems: Proximal algorithms, forward-backward splitting, and regularized Gauss-Seidel methods," *Convergence*, pp. 1–35, 2010.

[22] J. Portilla, "Image restoration through l0 analysis-based sparse optimization in tight frames," in *Proc. 16th IEEE ICIP*, 2009, pp. 3865–3868.

[23] T. Goldstein and S. Osher, "The split Bregman method for l1-regularized problems," *SIAM J. Imaging Sci.*, vol. 2, no. 2, pp. 323–343, Apr. 2009.

[24] J.-F. Cai, S. Osher, and Z. Shen, "Linearized Bregman iterations for frame-based image deblurring," *SIAM J. Imaging Sci.*, vol. 2, no. 1, pp. 226–252, Jan. 2009.

[25] J. Nocedal and S. Wright, *Numerical Optimization*. New York: Springer-Verlag, 2006.

[26] R. Pan and S. Reeves, "Efficient Huber-Markov edge-preserving image restoration," *IEEE Trans. Image Process.*, vol. 15, no. 12, pp. 3728–3735, Dec. 2006.

[27] N. Kingsbury, "Complex wavelets for shift invariant analysis and filtering of signals," *Appl. Comput. Harmon. Anal.*, vol. 10, no. 3, pp. 234–253, May 2001.

[28] M. Šorel and F. Šroubek, "Space-variant deblurring using one blurred and one underexposed image," in *Proc. Int. Conf. Image Process.*, 2009, pp. 157–160.