Robust Multichannel Blind Deconvolution via Fast Alternating Minimization

Filip Šroubek, *Member, IEEE*, and Peyman Milanfar, *Fellow, IEEE*

***Abstract*—Blind deconvolution, which comprises simultaneous blur and image estimations, is a strongly ill-posed problem. It is by now well known that if multiple images of the same scene are acquired, this multichannel (MC) blind deconvolution problem is better posed and allows blur estimation directly from the degraded images. We improve the MC idea by adding robustness to noise and stability in the case of large blurs or if the blur size is vastly overestimated. We formulate blind deconvolution as an ℓ_1-regularized optimization problem and seek a solution by alternately optimizing with respect to the image and with respect to blurs. Each optimization step is converted to a constrained problem by variable splitting and then is addressed with an augmented Lagrangian method, which permits simple and fast implementation in the Fourier domain. The rapid convergence of the proposed method is illustrated on synthetically blurred data. Applicability is also demonstrated on the deconvolution of real photos taken by a digital camera.**

***Index Terms*—Alternating minimization, augmented Lagrangian, blind deconvolution.**

I. Introduction

IMAGE deconvolution is a classical inverse problem in image processing. Deconvolution appears in a wide range of application areas, such as photography, astronomy, medical imaging, and remote sensing, just to name few. Images deteriorate during acquisition as data pass through the sensing, transmission, and recording processes. In general, the observed degradation is a result of two physical phenomena. The first is of random nature and appears in images as noise. The second is deterministic and results in blurring, which is typically modeled by convolution with some blur kernel called the point spread function (PSF). Degradation caused by convolution can thus appear in any application where image acquisition takes place. The common sources of blurring are lens imperfections, air turbulence, or camera-scene motion. Solving the deconvolution problem in a reliable way has been of prime interest in the field of image processing for several decades and has produced an enormous number of publications.

Let us first consider problems with just one degraded image, i.e., single-channel deconvolution. The simplest case is if the blur kernel is known (i.e., a classical deconvolution problem). However, even here, estimating an unknown image is ill-posed due to the ill-conditioned nature of the convolution operators. This inverse problem can only be solved by adopting some sort of regularization (in stochastic terms, regularization corresponds to priors). Another option is to use techniques such as coded aperture [1], but this requires a modification of camera hardware, which we do not consider here. A popular recent approach is to let the unknown image be represented as a linear combination of few elements of some frame (usually an overcomplete dictionary) and to force this sparse representation by using the ℓ_p norm $(0 \leq p \leq 1)$. Either we can search for the solution in the transform domain (coefficients of the frame elements), which is referred to as the synthesis approach, or regularize directly the unknown image, which is called the analysis approach. Analysis versus synthesis approach has been studied earlier [2], [3]. If the frame is an orthonormal basis, both approaches are equivalent. More interesting however is the case of redundant representation (e.g., an undecimated wavelet transform), when the two approaches differ. Conclusions presented in [3] suggest that, for deconvolution problems, the analysis approach is preferable because sparsity should be enforced only on a part of the redundant representation (e.g., high-pass bands), and this can be easily implemented only in the analysis approach. Very recently, it has been shown that the analysis approach is solved efficiently using variable splitting and by applying a Bregman iterative method [4] or an augmented Lagrangian method (ALM) [5] (both methods lead to the same algorithm).

If the blur kernel is unknown, we face single-channel blind deconvolution, which is clearly even more complicated than the classical deconvolution problem. This inverse problem is underdetermined as we have more unknowns (image and blur) than equations. For a long time, the problem seemed too difficult to solve for general blur kernels. Past algorithms usually worked only for special cases, such as astronomical images with a uniform (black) background, and their performance depended on initial estimates of PSFs. To name a few papers from this category, consider [6]–[8] and survey [9]. Probably, the first attempt toward a more general blur estimation came from Fergus *et al.* [10], who proposed a variational Bayesian method [11] with natural image statistics. This triggered a furious activity in the computer vision community, and soon, several conference papers appeared on the same topic [12]–[17]. Levin *et al.* [15]

Manuscript received February 28, 2011; revised July 28, 2011 and October 24, 2011; accepted November 01, 2011. Date of publication November 09, 2011; date of current version March 21, 2012. This work was supported in part by the US Air Force under Grant FA9550-07-1-0365, by the National Science Foundation under Grant CCF-1016018, by the Czech Ministry of Education under Project 1M0572 (Research Center DAR), and by the Grant Agency of the Czech Republic under Project P103/11/1552. F. Sroubek performed the work while at the UCSC supported by the Fulbright Visit Scholar Fellowship. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Ramin Samadani.

F. Šroubek is with the Institute of Information Theory and Automation, Academy of Sciences of the Czech Republic, 182 08 Prague, Czech Republic (e-mail: sroubekf@utia.cz).

P. Milanfar is with the Department of Electrical Engineering, University of California at Santa Cruz, Santa Cruz, CA 95064 USA (e-mail: milanfar@ee.ucsc.edu).

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.2011.2175740

pointed out that the joint posterior probability of the image–blur pair favors a trivial solution of the blur being a delta function and that marginalizing the posterior (integrating out the image variable) is more appropriate. However, a closed-form solution seldom exists, and a complicated approximation of the posterior is necessary, which leads to cumbersome methods that can hardly handle large blurs. In order to avoid these drawbacks, recent methods still try to minimize directly the joint posterior probability since it can be done in an efficient way but perform all sorts of tricks to avoid the trivial solution. Jia [12] uses an alpha matte to extract a transparency map and estimates the blur kernel on the map. Joshi *et al.* [13] predicts sharp edges using edge profiles and estimates the blur kernel from the predicted edges. Cho *et al.* [16] applies a shock filter and gradient thresholding to restore only strong edges and estimates the blur kernel from the truncated gradient image. A similar idea further improved by a kernel refinement step has been proposed recently by Xu *et al.* [17]. In general, the single-channel blind deconvolution methods get trapped in local minima and must estimate blurs using a multiscale approach. They have many parameters that influence the result considerably and are hard to tune. The common trick for the methods to work is to have means to predict strong edges. However, if the blurry image does not have salient edges or it is corrupted by noise, all the single-channel deconvolution methods usually fail.

The ill-posed nature of blind deconvolution can be remedied to a great extent by considering multiple images. In this case, the problem is referred to as multichannel (MC) blind deconvolution and will be the subject of our investigation. Acquired images must capture the same scene and differ only in the blur kernel. This may not seem to be easy to achieve in practice. However, the opposite is true. There are many situations where multiple images blurred in a slightly different way can be obtained. For example, if atmospheric turbulence causes blurring, we can capture several images (or video frames) in a row, and due to the random nature of turbulence, each image is almost surely blurred in a different way. If camera shake causes blurring, continuous shooting (or video capture) with the camera provides several images that are blurred in a different way since our hand moves randomly. MC deconvolution requires that the input images are properly registered, which is one drawback compared with the single-channel case. If the images are acquired as described above, misregistration is only minor, and even simple registration methods will provide accurate and stable results (see, e.g., [18]) for a survey of registration methods. We will thus assume that the input images are registered up to some global translation. A simple registration method for affine transforms is used in our experiments, as sketched in Section VI. More problematic is the occurrence of space-variant blur, which often arises in practice, such as rotating camera or profound depth of scene. We note that the method proposed here assumes a space-invariant case, but by applying the method locally, we can, in theory, deal with space-variant cases as well. We refer the interested reader to [19] and references therein for space-variant deconvolution.

One of the earliest intrinsic MC blind deconvolution methods [20] was designed particularly for images blurred by atmospheric turbulence. Harikumar *et al.* [21] proposed an indirect algorithm, which first estimates blur kernels and then recovers the original image by a standard nonblind method. The blur kernels are equal to the minimum eigenvector of a special matrix constructed from the blurred input images. Necessary assumptions for perfect recovery of the blurs are noise-free environment and channel coprimeness, i.e., a scalar constant is the only common factor of the blurs. Giannakis *et al.* [22] developed another indirect algorithm based on Bezout's identity of coprime polynomials, which finds restoration filters. In addition, by convolving the filters with the input images, it recovers the original image. Both algorithms are vulnerable to noise and, even for a moderate noise-level restoration, may break down. Pai *et al.* [23] suggested two MC restoration algorithms that, contrary to the previous two indirect algorithms, estimate directly the original image from the null space or from the range of a special matrix. Another direct method based on the greatest common divisor was proposed in [24]. In noisy cases, the direct algorithms are more stable than the indirect ones. Approaches based on the autoregressive moving average model are given in [25]. MC blind deconvolution using a Bussgang algorithm was proposed in [26], which performs well on spatially uncorrelated data, such as binary text images and spiky images. Sroubek *et al.* [27] proposed a method that reformulates Harikumar's idea in [21] as a MC regularization term and simultaneously minimizes an energy function with respect to the image and blur kernels. This allows us to handle inexact PSF sizes and to compensate for small misalignment in input images, which made MC deconvolution more practical. However, small PSFs (less than 15×15) and images of size couple of hundreds of pixels were only considered. It is mainly because of the inefficiency of the applied numerical algorithm that the method is not converging for larger blurs and images.

Here, we propose an MC blind deconvolution method that can handle very large blurs (e.g., 50×50) and images of several megapixels with even better accuracy and speed. The method is based on the same idea as in [27], and it is formulated as a constrained optimization problem. For image regularization, we use total variation (TV) [28], and for blur regularization, we use the MC constraint proposed in [21]. We show that the original MC constraint is not robust to noise and propose a simple remedy, which requires a negligible extra computation but achieves much better stability with respect to noise. Since the optimization problem mixes the ℓ_2 and ℓ_1 norms, we use the state-of-the-art numerical method of augmented Lagrangian [5] to solve the blind deconvolution problem and achieve very fast convergence. As it will be cleared later, positivity of blur kernels is an important constraint that must be included in the optimization problem. We show that positivity can be incorporated in augmented Lagrangian effortlessly without affecting the convergence properties.

This paper is organized as follows. Section II defines notation and presents the basic alternating minimization approach to blind deconvolution. Image regularization in the form of isotropic TV is given in Section III. Section IV discusses the problem of blur estimation in the MC scenario and influence of noise and blur size and proposes a novel blur kernel constraint with sparsity and positivity regularization. A description of the proposed algorithm is given in Section V, together with

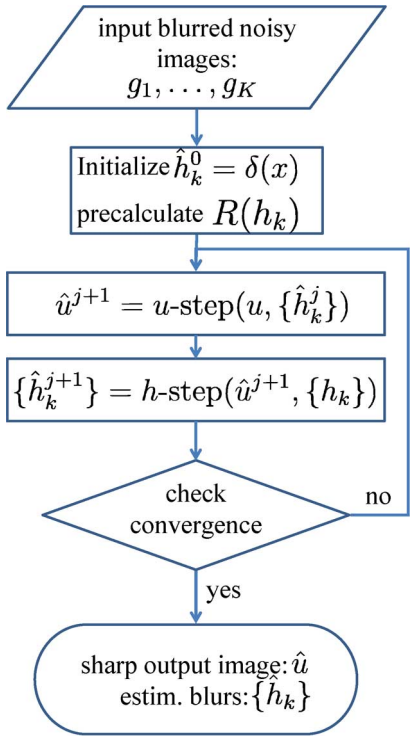


Fig. 1. Flowchart of the alternating minimization algorithm.

implementation details. The experimental section, Section VI, empirically validates the proposed method, and Section VII concludes this paper.

II. MC Blind Deconvolution Basics

We formulate the problem in the discrete domain and use frequently vector–matrix notation throughout the text. Images and PSFs are denoted by small italic letters and their corresponding vectorial representations (lexicographically ordered pixels) are denoted by small bold letters. The MC blind deconvolution problem assumes that we have $K > 1$ input images $\{g_1, \ldots, g_K\}$ ($g_k : \mathbb{N}^2 \to \mathbb{R}$) that are related to an unknown image $u : \mathbb{N}^2 \to \mathbb{R}$ according to model

$$g_k(i) = (h_k * u)(i) + n_k(i), \qquad 1 \le k \le K \tag{1}$$

where h_k denotes an unknown blur (kernel or PSF = PSF) and n_k is the additive noise in the kth observation. Operator $*$ stands for convolution, and $i \in \mathbb{N}^2$. When no ambiguity arises, we drop multiindex i from the notation. In the vector–matrix notation, (1) becomes

$$\mathbf{g}_k = \mathbf{H}_k\mathbf{u} + \mathbf{n}_k = \mathbf{U}\mathbf{h}_k + \mathbf{n}_k \tag{2}$$

where matrices $\mathbf{H}_k$ and $\mathbf{U}$ perform convolution with h_k and u, respectively. To denote the ith element in the vector notation, we write $[\cdot]_i$, e.g., $u(i) = [\mathbf{u}]_i$. The size of images and blurs (matrices and vectors) will be discussed later when necessary.

In the case of multiple acquisitions, we cannot expect that input images are perfectly spatially aligned. One can model such misregistation by the geometric transformation $\mathbf{W}_k$ that precede blurring $\mathbf{H}_k$, i.e., $\mathbf{H}_k\mathbf{W}_k\mathbf{u}$. If $\mathbf{W}_k$ is invertible, then $\mathbf{H}_k\mathbf{W}_k = \mathbf{W}_k\mathbf{W}_k^{-1}\mathbf{H}_k\mathbf{W}_k = \mathbf{W}_k\widetilde{\mathbf{H}}_k$, where $\widetilde{\mathbf{H}}_k = \mathbf{W}_k^{-1}\mathbf{H}_k\mathbf{W}_k$. If $\mathbf{H}_k$ is a standard convolution with some PSF h_k and $\mathbf{W}_k$ is a linear geometric transformation, then the new blurring operator $\widetilde{\mathbf{H}}_k$ remains a standard convolution but with h_k warped according to $\mathbf{W}_k$. Therefore, for linear geometric transformations (such as affine), the order of geometric transformation and blurring can be interchanged. We thus assume that input images g_k can be accurately registered by linear transformations, and a registration step preceding blind deconvolution removes such geometric transformations.

It is well known that the problem of estimating u from g_k is ill-posed; thus, this inverse problem can only be solved satisfactorily by adopting some sort of regularization. Formally, this leads to the following optimization problem:

$$\min_{u,\{h_k\}} F(u, \{h_k\}) + Q(u) + R(\{h_k\}) \tag{3}$$

where F is the data fidelity term and Q and R are regularizers of the image and blurs, respectively. The formation model (1) determines the data term leading to a standard formulation $F(u, \{h_k\}) = (\gamma/2)\sum_{k=1}^{K} \|u * h_k - g_k\|^2$, where γ is inversely proportional to the variance of noise n_k and $\|\cdot\|$ denotes the ℓ_2 norm. For simplicity, we assume the same noise variance in all frames; therefore, single parameter γ suffices. The standard approach to solve (3) is called alternating minimization and will be adopted here as well. We split the problem into two subproblems, i.e.,

$$\text{“}u\text{-step"}: \quad \min_{u} F(u, \{h_k\}) + Q(u) \tag{4}$$

$$\text{“}h\text{-step"}: \quad \min_{\{h_k\}} F(u, \{h_k\}) + R(\{h_k\}) \tag{5}$$

and alternate between them (see the algorithm flowchart in Fig. 1). Convergence to the global minimum is theoretically not guaranteed since the unknown variables are coupled in the data term F. However, we show that each subproblem separately converges to its global minimum and that it can be solved efficiently by the ALM. This implies that, in general, the global minimum of (3) is attainable after few alternations between the subproblems. The next two sections describe in detail the image Q and blur R regularization terms.

III. Image Regularization

A popular recent approach to image regularization is to assume that the unknown image u is represented as a linear combination of few elements of some frame (usually an overcomplete dictionary) and to force this sparse representation by using the ℓ_1 norm (or ℓ_0). Arguably, the best known and most commonly used image regularizer, which belongs to the category of sparse priors, is the TV norm [28].

The isotropic TV model is the ℓ_1 norm of image-gradient magnitude values and takes the following form:

$$Q(u) = \sum_i \phi(\nabla u(i)) = \sum_i \sqrt{(\nabla_x u(i))^2 + (\nabla_y u(i))^2} \tag{6}$$

where $\phi(x) = \|x\|$. The TV regularizer thus forces the solution to have sparse image gradient. Depending on the type of data, one can have sparsity in different domains. This modification is however easy to achieve. All we have to do is to replace

derivatives with a transformation (e.g., a waveletlike multiscale transform), which gives sparse representation of our data.

Using the vector–matrix notation, the isotropic TV (6) can be written as

$$Q(\mathbf{u}) = \Phi(\mathbf{D}_x\mathbf{u}, \mathbf{D}_y\mathbf{u}) = \sum_i \sqrt{[\mathbf{D}_x\mathbf{u}]_i^2 + [\mathbf{D}_y\mathbf{u}]_i^2} \quad (7)$$

where $\mathbf{D}_x$ and $\mathbf{D}_y$ are matrices performing derivatives with respect to x and y, respectively.

IV. Blur Estimation and Regularization

We first review an MC PSF estimation method proposed in [21], [22], which was later used in MC blind deconvolution as the PSF regularizer [27]. We demonstrate that the method is not robust to noise and show a novel improvement in this aspect. To keep the notation simple, let us assume 1-D data and the two-channel convolution model (1) ($K = 2$). The following discussion can be easily extended to 2-D data and any $K > 2$. The sizes of 1-D data g_k, u, and h_k is M, N, and L, respectively, with $N \gg L$. Noise n_k is of the same size as g_k. Kernels h_k can be of different sizes, but we can always pad the smaller ones with zeros to have the size of the largest one and therefore L refers to the size of the largest PSF. To deal correctly with convolution at image boundaries, we work with convolution that returns a "valid" part of the support and thus $M = N - L + 1$. The matrices $\mathbf{H}_k$ and $\mathbf{U}$ in the vector–matrix formation model (2) are thus of size $M \times N$ and $M \times L$, respectively.

Let $\widehat{h}_k$ be an estimate of h_k. In general, the original PSF size L is not known; therefore, $\widehat{h}_k$ can be of different size, which is denoted here as $\widehat{L}$. Let us study three cases that will be used in the following discussion: (a1) noiseless case ($n_k = 0$); (a2) PSF size is exactly known ($\widehat{L} = L$); and (a3) original PSFs are weakly coprime and images g_k are persistently exciting for size L. A set of kernels $\{h_k\}$ is called weakly coprime [22]; if there exists kernel s and set $\{\widetilde{h}_k\}$ so that, $\forall k$, $h_k = s * \widetilde{h}_k$, then s is a scalar. In other words, if the kernels are decomposable, they must not have a common kernel. An image z of size M is called persistently exciting [21] for size L if its "valid" convolution matrix $\mathbf{Z}$ of size $(M - L + 1) \times L$ has full column rank. Note that such an image will be also persistently exciting for any size smaller than L.

A. Noiseless Case

We first consider a situation, when all three assumptions (a1), a(2), and (a3) hold. If $\widehat{h}_k = h_k$, then

$$g_2 * \widehat{h}_1 - g_1 * \widehat{h}_2 = h_2 * u * \widehat{h}_1 - h_1 * u * \widehat{h}_2 = 0 \quad (8)$$

where we used the commutative property of convolution. Rewriting the above relation in the vector–matrix notation, we get

$$[\mathbf{G}_2, -\mathbf{G}_1]\mathbf{h} = \mathbf{0} \quad (9)$$

where $\mathbf{h} = [\mathbf{h}_1^T, \mathbf{h}_2^T]^T$. Matrices $\mathbf{G}_1$ and $\mathbf{G}_2$ denote "valid" convolution with g_1 and g_2, respectively, and they are of size $(M - L + 1) \times L$. Note that, in the case of $K > 2$, it is sufficient to consider all unordered pairs of images, which is equal to the combinatorial number $\binom{K}{2}$. Thus, for example, for $K = 3$, the number of image pairs is $\binom{3}{2} = 3$; (9) becomes

$$\begin{bmatrix} \mathbf{G}_2 & -\mathbf{G}_1 & \mathbf{0} \\ \mathbf{G}_3 & \mathbf{0} & -\mathbf{G}_1 \\ \mathbf{0} & \mathbf{G}_3 & -\mathbf{G}_2 \end{bmatrix} \mathbf{h} = \mathbf{0}.$$

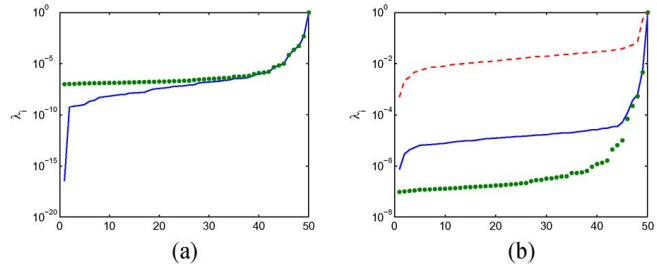


Fig. 2. Spectra of kernel regularization matrices $\mathbf{R}$ in (10), $\mathbf{R}_\Sigma$ in (15), and $\mathbf{R}_\Delta$ in (17). (a) $\mathbf{R}$ in the (solid line) noiseless and (dotted line) noisy case and (b) (solid line) $\mathbf{R}_\Sigma$ and (dashed line) $\mathbf{R}_\Delta$ in the noisy case.

Let us continue with $K = 2$ and define a symmetric positive semidefinite $2L \times 2L$ matrix, i.e.,

$$\mathbf{R} = [\mathbf{G}_2, -\mathbf{G}_1]^T[\mathbf{G}_2, -\mathbf{G}_1]. \quad (10)$$

The computational complexity of constructing this matrix is discussed in Section V-C. It follows from (9) that the correct estimates of h_k lie in the null space of $\mathbf{R}$. We refer to eigenvalues of $\mathbf{R}$ as λ_i ($\lambda_1 < \lambda_2 < \ldots < \lambda_{2L}$) and the corresponding eigenvectors as $\mathbf{v}_i$. Since (a2) and (a3) hold, $\mathbf{R}$ has exactly one zero eigenvalue λ_1, and eigenvector $\mathbf{v}_1$ is equal to the correct PSFs h_k stacked in one vector multiplied by a scalar. Note that $\mathbf{R}$ is constructed solely from the input image $\mathbf{g}_k$ values, and it can be thus used for the PSF estimation. An example of the $\mathbf{R}$ spectrum (plot of λ_i values) is in Fig. 2(a) (solid line). Matrix $\mathbf{R}$ was constructed from images blurred by two 5×5 PSFs in Fig. 3(a). Notice the prominent kink at the first eigenvalue λ_1. The corresponding eigenvector $\mathbf{v}_1$ represents exactly the original PSFs. This fact is also illustrated in Fig. 4(a), which plots the representation $\{\alpha_i\}$ of $\mathbf{h}$ in basis $\{\mathbf{v}_i\}$, i.e., $\mathbf{h} = \sum_{i=1}^{2L} \alpha_i \mathbf{v}_i$. One can use $\mathbf{R}$ to build the following quadratic form:

$$R(\mathbf{h}) = \mathbf{h}^T\mathbf{R}\mathbf{h} \quad (11)$$

and rewrite the $\mathbf{v}_1$ eigenvector estimation as a constrained optimization problem

$$\min_{\mathbf{h}} R(\mathbf{h}) \quad \text{s.t.} \quad \forall k \sum_i h_k(i) = 1. \quad (12)$$

As proposed in [27], it is better to use the quadratic term R as a PSF regularization term in the blind MC deconvolution problem (3). Because of the favorable spectrum of $\mathbf{R}$, the convergence of such algorithms is very fast.

B. Noisy Case

Let us see what happens if we remove (a1) and allow noise to enter the formation model (1). We assume uncorrelated normally distributed noise $n_k \sim N(0, \sigma^2)$. It follows from (2) that the convolution matrices $\mathbf{G}_k$ in (9) take the form

$$\mathbf{G}_k = \mathbf{H}_k\mathbf{U} + \mathbf{N}_k \quad (13)$$

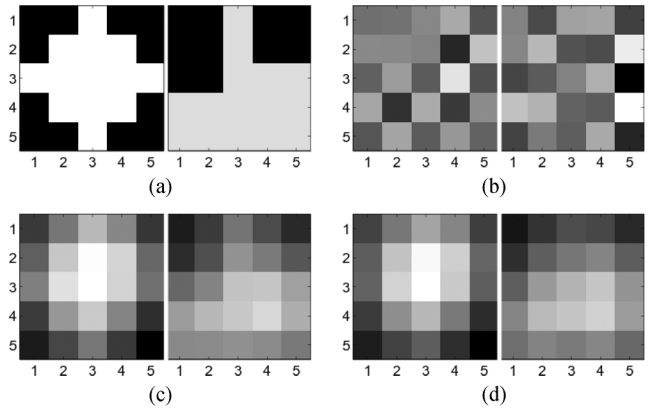


Fig. 3. PSFs and their estimates (first eigenvectors) in the noisy case. (a) two original PSFs of size 5×5. (b) Estimation using $\mathbf{R}$. (c) Estimation using $\mathbf{R}_\Sigma$. (d) Estimation using $\mathbf{R}_\Delta$.

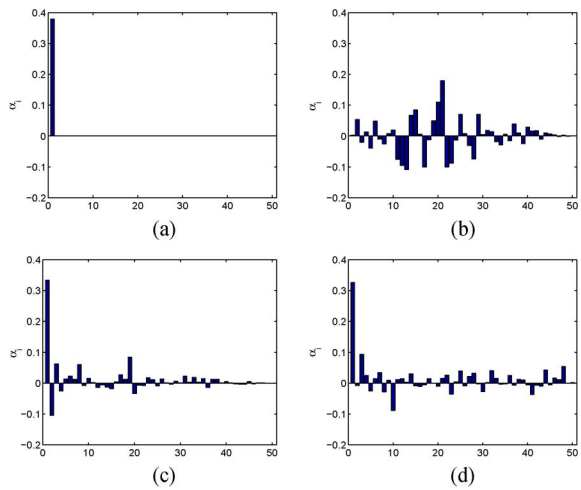


Fig. 4. Representation of PSFs in the eigenvector basis of regularization matrices. (a) $\mathbf{R}$ in the noiseless case. (b) $\mathbf{R}$ in the noisy case. (c) $\mathbf{R}_\Sigma$ in the noisy case. (d) $\mathbf{R}_\Delta$ in the noisy case.

where, this time, $\mathbf{H}_k$ is of size $(M-L)\times M$ and $\mathbf{N}_k$ is a noise convolution matrix constructed in the same way as $\mathbf{G}_k$ but using elements of n_k instead of g_k. Substituting for $\mathbf{G}_k$ in (9), we get

$$[\mathbf{G}_2, -\mathbf{G}_1]\mathbf{h} \sim N(\mathbf{0}, \Sigma) \tag{14}$$

where $\Sigma = \mathbf{cov}(\mathbf{H}_2\mathbf{U}\mathbf{h}_1 + \mathbf{N}_2\mathbf{h}_1 - \mathbf{H}_1\mathbf{U}\mathbf{h}_2 - \mathbf{N}_1\mathbf{h}_2) = \mathbf{cov}(\mathbf{N}_2\mathbf{h}_1 - \mathbf{N}_1\mathbf{h}_2) = \mathbf{cov}(\mathbf{H}_1\mathbf{n}_2 - \mathbf{H}_2\mathbf{n}_1) = \mathbf{H}_1\mathbf{cov}(\mathbf{n}_2)\mathbf{H}_1^T + \mathbf{H}_2\mathbf{cov}(\mathbf{n}_1)\mathbf{H}_2^T = \sigma^2(\mathbf{H}_1\mathbf{H}_1^T + \mathbf{H}_2\mathbf{H}_2^T)$, since $h_2 * u * h_1 = h_1 * u * h_2 \Longleftrightarrow \mathbf{H}_2\mathbf{U}\mathbf{h}_1 = \mathbf{H}_1\mathbf{U}\mathbf{h}_2$, which follows from (8). Because of noise, we cannot expect that the smallest eigenvalue of $\mathbf{R}$ will no longer be zero. Indeed, the kink visible in the noiseless case is completely leveled out in the noisy case. Fig. 2(a) (dotted line) shows the spectrum of $\mathbf{R}$ for the input data used before but corrupted by noise with SNR $= 40$ dB, which is a relatively small level of noise hardly detectable by human eyes. Eigenvector $\mathbf{v}_1$ is no longer informative and represents an erroneous solution, as shown in Fig. 3(b). The correct solution is a linear combination of all eigenvectors with the weights almost randomly distributed, as shown in Fig. 4(b).

The maximum-likelihood estimation of kernels must include the covariance matrix Σ in $\mathbf{R}$, i.e.,

$$\mathbf{R}_\Sigma = [\mathbf{G}_2, -\mathbf{G}_1]^T\Sigma^{-1}[\mathbf{G}_2, -\mathbf{G}_1]. \tag{15}$$

The spectrum of $\mathbf{R}_\Sigma$ retains the kink at the first smallest eigenvalue λ_1, as Fig. 2(b) (solid line) shows. For comparison, we show the original spectrum of $\mathbf{R}$ in (10), as a dotted line [also in Fig. 2(a)]. The eigenvector $\mathbf{v}_1$ of $\mathbf{R}_\Sigma$ captures the original PSFs, as shown in Fig. 3(c). Encoding of the true kernels $\mathbf{h}$ in the basis $\{\mathbf{v}_i\}$ is relatively sparse and cluster around the smallest eigenvalues [see Fig. 4(c)]. The same behavior persists even for much higher noise levels (around 10 dB). The construction of $\mathbf{R}_\Sigma$ has one severe drawback: We must know the correct kernels h_k *a priori* in order to build Σ. Since our aim is to estimate PSFs, this seem to be contradictory. One can apply an iterative procedure and update Σ with every new estimate of h_k, as proposed in [21]. Unfortunately, this framework is not guaranteed to converge. In addition, inversion of Σ can be very costly, which makes the whole calculation of $\mathbf{R}_\Sigma$ for large kernels (large L) impossible.

We propose to filter the blurred input images g_k in such a way so that $\mathbf{R}$ without Σ in (10) will be closed to $\mathbf{R}_\Sigma$ in (15). If we filter the input images with some kernel p, then

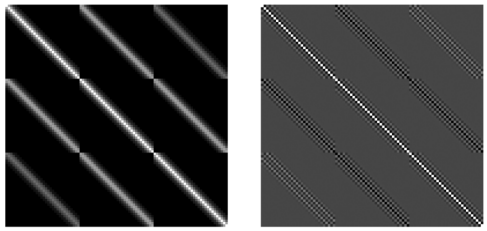
$$[\mathbf{P}\mathbf{G}_2, -\mathbf{P}\mathbf{G}_1]\mathbf{h} \sim N(\mathbf{0}, \Sigma_\mathbf{P}) \tag{16}$$

where $\mathbf{P}$ performs convolution with p and the covariance matrix is $\Sigma_\mathbf{P} = \sigma^2(\mathbf{H}_1\mathbf{P}\mathbf{P}^T\mathbf{H}_1^T + \mathbf{H}_2\mathbf{P}\mathbf{P}^T\mathbf{H}_2^T) = \sigma^2\mathbf{P}(\mathbf{H}_1\mathbf{H}_1^T + \mathbf{H}_2\mathbf{H}_2^T)\mathbf{P}^T$. The best choice of the filter p is such that $\Sigma_\mathbf{P} = \sigma^2\mathbf{I}$; since then, the covariance matrix can be neglected. However, this would again require *a priori* knowledge of unknown kernels h_k since p depends on h_k. Achieving a diagonal correlation matrix means that we want to spatially decorrelate the blur kernels. In the absence of any prior knowledge of the blurs, we wish to employ a decorrelation method that is sufficiently general. As such, given the well-accepted assumption of sparsity on high-frequency spatial structures, the natural choice is to apply a Laplacian operator. The justification is therefore empirical but quite reasonable. In Fig. 5(a), we show a small part of the covariance matrix Σ for our example with two blurs and, in Fig. 5(b), the covariance matrix $\Sigma_\mathbf{P}$ with $\mathbf{P}$ being the Laplacian. The covariance matrix of the filtered images is not diagonal but close to diagonal. The Laplacian produces images, which are relatively sparse and therefore spatially uncorrelated to a great extent. The same holds for PSFs that blur the images, which accounts for the close-to-diagonal covariance matrix.

Let Δ denote a matrix that performs convolution with the discrete Laplacian kernel l (in 1-D $l = [1, -2, 1]$). The proposed modification of the matrix $\mathbf{R}$ is

$$\mathbf{R}_\Delta = [\Delta\mathbf{G}_2, -\Delta\mathbf{G}_1]^T[\Delta\mathbf{G}_2, -\Delta\mathbf{G}_1]. \tag{17}$$

Matrix $\mathbf{R}_\Delta$ depends only on the input images g_k, and the construction is trivial. The spectrum of this matrix retains the kink [see dashed line in Fig. 2(b)] and relatively sparse representation of $\mathbf{h}$, as shown in Fig. 4(d). Eigenvector $\mathbf{v}_1$ estimates h_k in a similar way as ideal $\mathbf{R}_\Sigma$ [see Fig. 3(d)].



(a) (b)

Fig. 5. Covariance matrices. (a) Calculated from the original PSFs. (b) Calculated from the Laplacian of PSFs.

C. Overestimated Kernel Size

It is unrealistic to assume that the kernel size L is exactly known in practice. Let us thus consider the case when both (a1) and (a3) hold, but (a2) is violated with the kernel size being overestimated, i.e., $\widehat{L} > L$. We can readily see that if $\widehat{h}_k = s * h_k$, where s is an arbitrary spurious kernel of size $S = \widehat{L} - L + 1$, the MC constraint (8) still holds

$$g_2 * \widehat{h}_1 - g_1 * \widehat{h}_2 = h_2 * u * s * h_1 - h_1 * u * s * h_2 = 0. \quad (18)$$

In the language of matrix eigenvalues and eigenvectors, this fact translates as follows. Matrix $\mathbf{R}$ defined in (10) is of size $\widehat{L} \times \widehat{L}$. The correct kernels lie again in the null space of $\mathbf{R}$, but this time, the matrix nullity is of the size of the spurious kernel, i.e., nullity$(\mathbf{R}) = S$. The regularization term (11) built from $\mathbf{R}$ becomes less restrictive (more "flat") because of the increased nullity. Therefore, convergence of any minimization algorithm, which estimates PSFs using the proposed regularizer R, is seriously hindered in the case of overestimated kernel size. Note that if the kernel size $\widehat{L}$ is underestimated, (18) does not hold, and we cannot estimated the kernels at all. We will not consider the underestimated case and, instead, focus on improving the stability of the overestimated case.

One can be tempted to assume that the unconstrained optimization problem, as defined in (5), would eliminate the ambiguity inherent in $R(\mathbf{h})$. Using the vector–matrix notation, this problem rewrites as

$$\min_{\mathbf{h}} \frac{\gamma}{2} \sum_{k=1}^{2} \|\widehat{\mathbf{U}}\mathbf{h}_k - \mathbf{g}_k\|^2 + R(\mathbf{h}) \quad (19)$$

where $\widehat{\mathbf{U}}$ is the $M \times \widehat{L}$ convolution matrix with the estimate $\widehat{u}$ of the original image u. If estimate $\widehat{u} = u$, the above optimization problem is well posed, and in fact, we do not need regularizer R at all. However, this scenario is unrealistic since we do not know the original image. Alternating minimization often starts with $\widehat{u}$ equal to a so-called average image, i.e., $\widehat{u} = (1/K)\sum_k g_k$. To illustrate the behavior of the data term $F(\widehat{u}, \{\widehat{h}_k\})$ with respect to the spurious kernel s, we conducted the following experiment. We generated two blurry signals g_1 and g_2 using some random positive PSFs h_1 and h_2 of size L. We set $\widehat{L} = L + 1$; therefore, the spurious kernel s is of size 2, and $s = [s_1, s_2]$. Let us consider kernels of form $\widehat{h}_k = s * h_k$ that preserve energy $\sum_i \widehat{h}_k(i) = 1$, then $R(\widehat{\mathbf{h}}) = 0$ for any s and $s_2 = 1 - s_1$. The data term $F(\widehat{u}, \widehat{h}_k)$ with $\widehat{u}$ being the average image is a function of s_1, and we plot its values for different s_1 in Fig. 6.

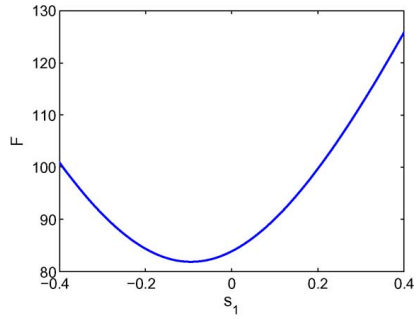


Fig. 6. Data term $F(\widehat{u}, \{s * h_k\})$ as a function of the first elements s_1 of the 2×1 spurious vector $s = [s_1, s_2]$, where $s_2 = 1 - s_1$. The minimum is not reached for $s = [0, 1]$ (delta function) but for s with a small negative value.

The minimum was reached for a negative value of s_1, and the same behavior was observed for any pair of blurs h_1 and h_2. The data term is thus biased toward kernels with small negative values, and the unconstrained optimization problem (19) is inappropriate if the kernel size is overestimated. An intuitive explanation is the following. Since we use the average image, the value of F would reach its minimum for some $\widehat{h}_k$ close to delta functions. Such a solution is however heavily penalized by R, which allows only PSFs of form $s * h_k$. In order to get closer to the delta-function solution, s must act as an inverse filter to all positive h_k, and this means that it must perform differentiation; hence, negative values in s are inevitable.

Forcing positivity on kernels is the remedy to the above problem. Clearly, this approach is possible only for positive kernels. We encounter positive-only kernels in many deconvolution problems, and making this assumption is thus not very restrictive. With the positivity constraint, the above problem can be solved by means of quadratic programming. Here, we show a different approach, which will allow us an elegant integration in the ALM and much faster implementation than quadratic programming. We have empirically observed that forcing sparsity on h_k further boosts convergence. In order to guarantee both positivity and sparsity, we propose to use a new kernel regularizer, i.e.,

$$R(\mathbf{h}) = \frac{\delta}{2}\mathbf{h}^T\mathbf{R}_\Delta\mathbf{h} + \Psi(\mathbf{h}) \quad (20)$$

where

$$\Psi(\mathbf{h}) = \sum_{k=1}^{K}\sum_{i=1}^{\widehat{L}} \psi\left(h_k(i)\right), \quad \psi(t) = \begin{cases} t, & \text{if } t \geq 0 \\ +\infty, & \text{otherwise} \end{cases} \quad (21)$$

and δ is the weight that controls the influence of the MC constraint $\mathbf{R}_\Delta$. The definition of ψ ensures positivity by absolutely penalizing negative values and forces sparsity by calculating the ℓ_1 norm of positive kernels.

Note that it is not necessary to explicitly include the constraint $\forall k \sum_i h_k(i) = 1$ as in (12), which preserves the average gray value in images. This constraint is automatically enforced by the fidelity term $\sum_{k=1}^{2} \|u * h_k - g_k\|$ in (19). If the mean value of the estimated image u is equal to the mean value of g_k, then by solving (19) (h-step), we always preserve $\sum_i h_k(i) = 1$. The u-step in (4) does not change the mean value of u either because the fidelity term is present there as well. Therefore, the

condition is not modified in alternating minimization, and we only have to guarantee that initial PSFs follow the constraint.

D. Kernel Coprimeness

Let us consider the assumption (a3) of persistently exciting images and weakly coprime kernels. The condition of persistently exciting image is a very mild one. Usually $M \gg L$, convolution matrices $\mathbf{G}_k$ have many more rows than columns, and the probability that the matrices will not have a full column rank is thus very small. We do not consider here degenerate cases, such as perfectly uniform or periodic images, that may not be persistently exciting.

The condition of weakly coprime kernels may seem to be more problematic. In the 1-D case (signals), any kernel of length L can be decomposed (factorized) into $L-1$ kernels (root factors) of size 2, which is the direct consequence of the fundamental theorem of algebra[1] (see, e.g., [29]). It is therefore likely that there might exist a factor common to all kernels h_k. In the 2-D case (images), no such factorization in general exists and, as also discussed in [21], coprimeness holds deterministically for most of the 2-D cases of practical interest.

If the common factor exists despite its low probability, kernel estimation still partially works. We are able to recover kernels without their common factor, and the common factor remains as a blur in the estimated image.

V. Optimization Algorithm

Alternating minimization, which solves the MC blind deconvolution problem (3), consists of two subproblems: minimization with respect to the image (u-step) and the minimization with respect to the blurs (h-step). Both subproblems share some similarities because both the image (7) and the blur regularizer (20) are not smooth and introduce nonlinearity in the problem. Direct minimization in each step would be thus a slow process. A simple procedure that solves such problems is called variable splitting, which decouples the ℓ_2 and ℓ_1 portions of the problem (3) by introducing auxiliary variable and converting each subproblem to two simpler minimization steps. We then apply the ALM, which is equivalent to the split Bregman iterative method [4], to solve the subproblems. Our derivation follows the work presented in [5] and partially in [4]. Unique aspects of our algorithm will be emphasized. From now on, we will exclusively use the vector–matrix notation and stack all observations into one system by using the compact notations $\mathbf{g} = [\mathbf{g}_1^T, \ldots, \mathbf{g}_K^T]^T$, $\mathbf{h} = [\mathbf{h}_1^T, \ldots, \mathbf{h}_K^T]^T$, $\mathbf{H} = [\mathbf{H}_1^T, \ldots, \mathbf{H}_K^T]^T$, and the convolution matrix $\mathbf{U}$ will now denote a block diagonal matrix with K blocks, where each block is the original $\mathbf{U}$ from (2).

A. U-Step

Using the TV regularizer (7), minimization with respect to the image (4) writes as

$$\min_{\mathbf{u}} \frac{\gamma}{2}\|\mathbf{H}\mathbf{u}-\mathbf{g}\|^2 + \Phi(\mathbf{D}_x\mathbf{u}, \mathbf{D}_y\mathbf{u}). \tag{22}$$

[1]However, some of the factors may contain complex values.

Applying variable splitting, we replace $\mathbf{D}_x\mathbf{u}$ by $\mathbf{v}_x$ and $\mathbf{D}_y\mathbf{u}$ by $\mathbf{v}_y$. This yields a constrained problem

$$\min_{\mathbf{u},\mathbf{v}_x,\mathbf{v}_y} \frac{\gamma}{2}\|\mathbf{H}\mathbf{u}-\mathbf{g}\|^2 + \Phi(\mathbf{v}_x,\mathbf{v}_y) \quad \text{s.t.} \quad \mathbf{v}_x = \mathbf{D}_x\mathbf{u}, \mathbf{v}_y = \mathbf{D}_y\mathbf{u} \tag{23}$$

which is equivalent to (22). The ALM (or split-Bregman iteration) tackles the constrained problem (23) by considering the functional

$$\mathcal{L}_u(\mathbf{u},\mathbf{v}_x,\mathbf{v}_y) = \frac{\gamma}{2}\|\mathbf{H}\mathbf{u}-\mathbf{g}\|^2 + \Phi(\mathbf{v}_x,\mathbf{v}_y) + \frac{\alpha}{2}\|\mathbf{D}_x\mathbf{u}-\mathbf{v}_x-\mathbf{a}_x\|^2 + \frac{\alpha}{2}\|\mathbf{D}_y\mathbf{u}-\mathbf{v}_y-\mathbf{a}_y\|^2 \tag{24}$$

and solving it with an iterative algorithm:

Algorithm: $\widehat{\mathbf{u}} = u\text{-step}(\mathbf{u}^0)$

1: Set $\mathbf{v}_x^0 = \mathbf{v}_y^0 = \mathbf{a}_x^0 = \mathbf{a}_y^0 = 0$ and $j = 0$

2: **repeat**

3: $\mathbf{u}^{j+1} = \arg\min_{\mathbf{u}} \mathcal{L}_u(\mathbf{u}, \mathbf{v}_x^j, \mathbf{v}_y^j) \iff [\mathbf{H}^T\mathbf{H} + (\alpha/\gamma)(\mathbf{D}_x^T\mathbf{D}_x + \mathbf{D}_y^T\mathbf{D}_y)]\mathbf{u}^{j+1} = \mathbf{H}^T\mathbf{g} + (\alpha/\gamma)[\mathbf{D}_x^T(\mathbf{v}_x^j + \mathbf{a}_x^j) + \mathbf{D}_y^T(\mathbf{v}_y^j + \mathbf{a}_y^j)]$

4: $\{\mathbf{v}_x^{j+1}, \mathbf{v}_y^{j+1}\} = \arg\min_{\mathbf{v}_x,\mathbf{v}_x} \mathcal{L}_u(\mathbf{u}^{j+1}, \mathbf{v}_x, \mathbf{v}_y) \iff$
$[\mathbf{v}_x^{j+1}]_i = [\mathbf{D}_x\mathbf{u}^{j+1} - \mathbf{a}_x^j]_i[\mathbf{s}]_i^{-1}\max([\mathbf{s}]_i - (1/\alpha), 0)$,
$[\mathbf{v}_y^{j+1}]_i = [\mathbf{D}_y\mathbf{u}^{j+1} - \mathbf{a}_y^j]_i[\mathbf{s}]_i^{-1}\max([\mathbf{s}]_i - (1/\alpha), 0)$,

where

$[\mathbf{s}]_i = \sqrt{[\mathbf{D}_x\mathbf{u}^{j+1} - \mathbf{a}_x^j]_i^2 + [\mathbf{D}_y\mathbf{u}^{j+1} - \mathbf{a}_y^j]_i^2}$

5: $\mathbf{a}_x^{j+1} = \mathbf{a}_x^j - \mathbf{D}_x\mathbf{u}^{j+1} + \mathbf{v}_x^{j+1}$

$\mathbf{a}_y^{j+1} = \mathbf{a}_y^j - \mathbf{D}_y\mathbf{u}^{j+1} + \mathbf{v}_y^{j+1}$

6: $j \leftarrow j+1$

7: **until** stopping criterion is satisfied

8: **return** $\widehat{\mathbf{u}} \leftarrow \mathbf{u}^j$

This iterative algorithm consists of three update steps: lines 3, 4, and 5. Variables $\mathbf{a}_x$ and $\mathbf{a}_y$ are introduced by the ALM. Their update on line 5 is trivial. It is worth drawing a relation of the ALM to a penalty method. If we omit the updating step for $\mathbf{a}_x$ and $\mathbf{a}_y$, and keep $\mathbf{a}_x = \mathbf{a}_y = 0$, the above algorithm defaults to the penalty method. The penalty method converges to the solution of the constrained problem (23) only if we keep increasing α to infinity while iterating, as advocated in [30]. This is however not practical as the problem becomes gradually more ill-posed with increasing α. This drawback is avoided in the ALM. Since Φ is a lower semicontinuous proper convex function,[2] and $[\mathbf{D}_x^T, \mathbf{D}_y^T]^T$ has a full column rank, then, if (23) has a solution, the u-step algorithm converges to this solution even for α that is relatively small and fixed. This important theorem was proved in [31].

[2]In our case, Φ is continuous and thus lower semicontinuous

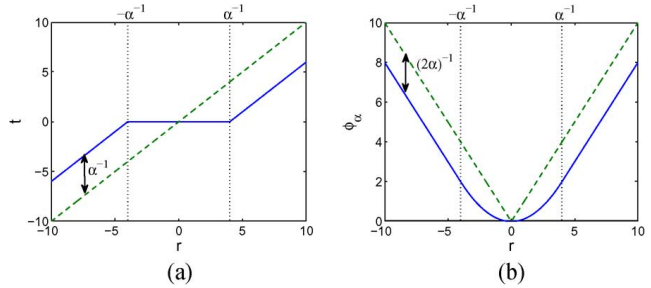


Fig. 7. Soft thresholding. (a) Shrinkage formula (26) for a nonzero threshold $1/\alpha$ (solid) and for $1/\alpha = 0$ (dashed). (b) Corresponding ϕ_α in (25) for a (solid) nonzero threshold $1/\alpha$ and (dashed) for $1/\alpha = 0$. Note that ϕ_α is a relaxed form of the ℓ_1 norm, which is the absolute value (dashed) in this simple case.

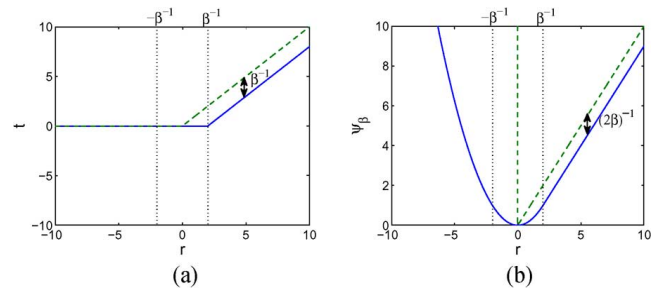


Fig. 8. Thresholding in the blur domain. (a) Shrinkage formula (32) for (solid) a nonzero threshold $1/\beta$ and for (dashed) $1/\beta = 0$. (b) Corresponding ψ_β in (31) for a nonzero threshold $1/\beta$ (solid) and for $1/\beta = 0$ (dashed).

Since $\mathcal{L}_u$ in (24) is quadratic with respect to $\mathbf{u}$, minimization on line 3 is a solution to a set of linear equations. We show later that this can be solved efficiently in the Fourier domain.

The beauty of variable splitting is that minimization with respect to $\mathbf{v}_x$ and $\mathbf{v}_y$ is, by definition, the Moreau proximal mapping [32] of Φ applied to $\mathbf{D}_x\mathbf{u}^{j+1} - \mathbf{a}_x^j$ and $\mathbf{D}_y\mathbf{u}^{j+1} - \mathbf{a}_y^j$. The problem can be solved for each ith element independently. Let $\mathbf{t} = [[\mathbf{v}_x]_i, [\mathbf{v}_y]_i]^T$ and $\mathbf{r} = [[\mathbf{D}_x\mathbf{u} - \mathbf{a}_x]_i, [\mathbf{D}_y\mathbf{u} - \mathbf{a}_y]_i]^T$ be vectors of size 2×1; the problem on line 4 is of the form

$$\phi_\alpha(\mathbf{r}) = \min_{\mathbf{t}} \left(\frac{\alpha}{2}\|\mathbf{r} - \mathbf{t}\|^2 + \|\mathbf{t}\| \right) \tag{25}$$

and, as proved in [30], the minimum is reached for

$$\mathbf{t} = \frac{\mathbf{r}}{\|\mathbf{r}\|} \max\left(\|\mathbf{r}\| - \frac{1}{\alpha}, 0 \right) \tag{26}$$

which is a generalized shrinkage formula for vectors. For the $\mathbf{r}$ scalar, (26) corresponds to a well-known soft-thresholding formula plotted as a solid line in Fig. 7(a). It is interesting to note that, after substituting for $\mathbf{t}$ in (25), $\phi_\alpha(\mathbf{r})$ [solid line in Fig. 7(b)] can be written in a closed form

$$\phi_\alpha(\mathbf{r}) = \begin{cases} \frac{\alpha}{2}\|\mathbf{r}\|^2, & \text{if } \|\mathbf{r}\| < \frac{1}{\alpha} \\ \|\mathbf{r}\| - \frac{1}{2\alpha}, & \text{otherwise} \end{cases} \tag{27}$$

which is a relaxed form of the original $\phi(\mathbf{r}) = \|\mathbf{r}\|$ in the isotropic TV definition (6). If $\alpha \to \infty$, then $\phi_\alpha \to \phi$, and the corresponding graphs are plotted as dashed lines in Fig. 7.

B. H-Step

The kernel estimation proceeds analogously to the u-step. Using the proposed regularizer (20), minimization with respect to the PSFs (5) writes as

$$\min_{\mathbf{h}} \frac{\gamma}{2}\|\mathbf{U}\mathbf{h} - \mathbf{g}\|^2 + \frac{\delta}{2}\mathbf{h}^T\mathbf{R}_\Delta\mathbf{h} + \Psi(\mathbf{h}) \tag{28}$$

Applying variable splitting $\mathbf{w} = \mathbf{h}$ yields the constrained problem

$$\min_{\mathbf{h},\mathbf{w}} \frac{\gamma}{2}\|\mathbf{U}\mathbf{h} - \mathbf{g}\|^2 + \frac{\delta}{2}\mathbf{h}^T\mathbf{R}_\Delta\mathbf{h} + \Psi(\mathbf{w}) \quad \text{s.t.} \quad \mathbf{w} = \mathbf{h} \tag{29}$$

Then, we consider the following functional:

$$\mathcal{L}_h(\mathbf{h},\mathbf{w}) = \frac{\gamma}{2}\|\mathbf{U}\mathbf{h} - \mathbf{g}\|^2 + \frac{\delta}{2}\mathbf{h}^T\mathbf{R}_\Delta\mathbf{h} + \Psi(\mathbf{w}) + \frac{\beta}{2}\|\mathbf{h} - \mathbf{w} - \mathbf{b}\|^2 \tag{30}$$

and solve it with the following iterative algorithm:

Algorithm: $\widehat{\mathbf{h}} = h\text{-step}(\mathbf{h}^0)$

1: Set $\mathbf{w}^0 = \mathbf{b}^0 = 0$ and $j = 0$

2: **repeat**

3: $\mathbf{h}^{j+1} = \arg\min_{\mathbf{h}} \mathcal{L}_h(\mathbf{h}, \mathbf{w}^j) \iff [\mathbf{U}^T\mathbf{U} + (\delta/\gamma)\mathbf{R}_\Delta + (\beta/\gamma)\mathbf{I}]\mathbf{h}^{j+1} = \mathbf{U}^T\mathbf{g} + (\beta/\gamma)(\mathbf{w}^j + \mathbf{b}^j)$

4: $\mathbf{w}^{j+1} = \arg\min_{\mathbf{w}} \mathcal{L}_h(\mathbf{h}^{j+1}, \mathbf{w}) \iff [\mathbf{w}^{j+1}]_i = \max([\mathbf{h}^{j+1} - \mathbf{b}^j]_i - (1/\beta), 0)$

5: $\mathbf{b}^{j+1} = \mathbf{b}^j - \mathbf{h}^{j+1} + \mathbf{w}^{j+1}$

6: $j \leftarrow j + 1$

7: **until** stopping criterion is satisfied

8: **return** $\widehat{\mathbf{h}} \leftarrow \mathbf{h}^j$

Matrix $\mathbf{I}$ denotes identity of size $K\widehat{L} \times K\widehat{L}$. As in the u-step, the h-step iterative algorithm consists of three update steps: lines 3, 4, and 5. Since $\mathcal{L}_h$ in (30) is quadratic with respect to $\mathbf{h}$, minimization on line 3 is a solution to a set of linear equations. This time, the minimization with respect to $\mathbf{w}$ is again the Moreau proximal mapping of Ψ applied to $\mathbf{h}^{j+1} - \mathbf{b}^j$, and it is solved elementwise. Let $t = [\mathbf{w}]_i$ and $r = [\mathbf{h} - \mathbf{b}]_i$; the problem on line 4 is of the following form:

$$\psi_\beta(r) = \min_t \left(\frac{\beta}{2}(r - t)^2 + \psi(t) \right) \tag{31}$$

where ψ is our positivity–sparsity enforcing function defined in (21) and plotted as dashed line in Fig. 8(b). After some manipulation, one can see that the minimum is reached for

$$t = \max\left(r - \frac{1}{\beta}, 0 \right). \tag{32}$$

The plot of this "one-sided" thresholding function is the solid line in Fig. 8(a). Using the thresholding function, a closed form of ψ_β is

$$\psi_\beta(r) = \begin{cases} \frac{\beta}{2}r^2, & \text{if } r < \frac{1}{\beta} \\ r - \frac{1}{2\beta}, & \text{otherwise} \end{cases} \tag{33}$$

with a plot in Fig. 8(b), i.e., the solid line. The function linearly increases in the positive domain, whereas in the negative domain, it increases quadratically. If $\beta \to \infty$, then $\psi_\beta \to \psi$, and the thresholding function in (32) approaches the dashed line in Fig. 8(a). However, as in the u-step, we do not need to increase β to infinity for the h-step algorithm to converge to the solution of the constrained problem (29). The ALM approach with its extravariable $\mathbf{b}$ converges. Note, that ψ must be a lower semicontinuous proper convex function for the method to converge, which is the case. Interestingly, if we replaced in definition (21) infinity with some large but finite numbers, the resulting function would no longer be convex. Infinity in the definition might look dangerous, but it turns out to give an elegant solution in the form of the thresholding function (32).

C. Implementation

We have analyzed the main points (u-step and h-step) of the optimization algorithm. Now we proceed with the description of the main loop of the algorithm and the computational cost of individual steps. Let N denote the number of pixels in the output image u, and let $\widehat{L}$ denote the number of pixels in our overestimated PSF support. The main loop of the MC blind deconvolution alternating minimization algorithm looks as follows:

MC blind deconvolution

Require: $K \geq 2$ input images $\{\mathbf{g}_k\}$; blur size $\widehat{L}$; parameters α, β, δ, γ

1: Set $j = 0$, $\widehat{\mathbf{h}}_k^0$'s to delta functions, and $\widehat{\mathbf{u}}^0 = 0$

2: Calculate $\mathbf{R}_\Delta$

3: **repeat**

4: $\widehat{\mathbf{u}}^{j+1} = u\text{-step}(\widehat{\mathbf{u}}^j, \widehat{\mathbf{h}}^j)$

5: $\widehat{\mathbf{h}}^{j+1} = h\text{-step}(\widehat{\mathbf{u}}^{j+1}, \widehat{\mathbf{h}}^j)$

6: $j \leftarrow j + 1$

7: **until** stopping criterion is satisfied

8: **return** $\widehat{\mathbf{u}} \leftarrow \widehat{\mathbf{u}}^j$

The stopping criterion, which we typically use, is $\|\widehat{\mathbf{h}}^j - \widehat{\mathbf{h}}^{j-1}\|/\|\widehat{\mathbf{h}}^j\| < tol$. The same can be used in the h-step and, likewise, in the u-step using $\mathbf{u}$ instead of $\mathbf{h}$. The calculation of $\mathbf{R}_\Delta$ can be done using the fast Fourier transform (FFT) without explicitly constructing the convolution matrices $\mathbf{G}_k$. Since $\mathbf{G}_k$ values are "valid" convolutions, we can construct only one row of $\mathbf{R}_\Delta$ at a time, and the overall complexity is thus $O(K\widehat{L}N \log N)$.

In general, the most time-consuming is the u-step, which requires an inversion of the huge $N \times N$ matrix $[\mathbf{H}^T\mathbf{H} + (\alpha/\gamma)(\mathbf{D}_x^T\mathbf{D}_x + \mathbf{D}_y^T\mathbf{D}_y)]$. One can apply iterative solvers, such as conjugate gradient, to avoid direct inversion, but we can do even better and have a direct solver. In our formulation, $\mathbf{H}$, $\mathbf{D}_x$, and $\mathbf{D}_y$ are convolution matrices. To avoid any ringing artifacts close to image boundaries, they should perform "valid" convolution, i.e., the output image is smaller and covers a region where both the input image and the convolution kernel are fully defined. If we properly adjust the image borders, e.g., by using the function edgetaper in MATLAB, we can replace "valid" convolution with block-circulant one, and ringing artifacts will be almost undetectable. The TV regularizer also helps to reduce such artifacts. FFT diagonalizes block-circulant convolution matrices, and inversion is thus straightforward. The remaining update steps for $\mathbf{v}_x$ ($\mathbf{v}_y$) and $\mathbf{a}_x$ ($\mathbf{a}_y$) are simple and can be computed in $O(N)$ time. The u-step is thus carried out with an overall $O(N \log N)$ cost.

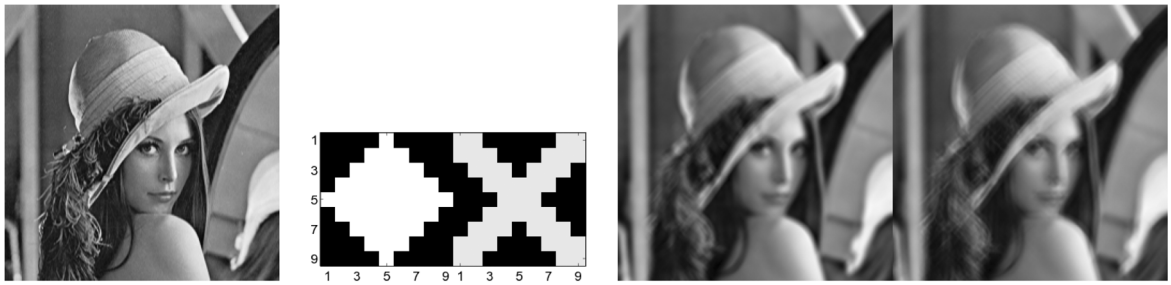
Unlike the u-step, which is calculated almost entirely in the Fourier domain, we perform the h-step in the image domain since we need the constrained kernel support $\widehat{L}$. Otherwise, $\mathbf{R}_\Delta$ becomes a very uninformative regularizer, as explained in Section IV-C. On line 3 of the h-step algorithm, we have to invert matrix $[\mathbf{U}^T\mathbf{U} + (\delta/\gamma)\mathbf{R}_\Delta + (\beta/\gamma)\mathbf{I}]$, which is of size $K\widehat{L} \times K\widehat{L}$ and thus much smaller than the matrix in the u-step. Typically, the size of blurs is not more than 40×40 pixels ($L = 1600$), and for two input images ($K = 2$), the matrix size is 3200×3200, which is still relatively small.[3] One can again apply an iterative solver such as a conjugate gradient, but we found it much more efficient to store the whole matrix and perform Cholesky decomposition to solve this problem. This can be computed in $O((K\widehat{L})^3)$ time. Again, update steps for $\mathbf{w}$ and $\mathbf{b}$ are very simple and require $O(K\widehat{L})$ operations.

Setting parameters is based solely on our empirical studies and cannot be considered as a rigorous procedure. The optimization method has four parameters. We have noticed that, in general, they can be fixed relative to one of them, i.e., γ, which depends on the noise level. This observation is not superficial. Afonso *et al.* [5] (as well as is [4] for the split Bregman method) also recommend to set parameters introduced by the ALM, i.e., in our case, α and β, with respect to the weight γ of the fidelity term. Parameter δ, which is the weight of the MC constraint term $\mathbf{h}^T\mathbf{R}_\Delta\mathbf{h}$, is proportional to the noise variance, as shown in (16), and therefore should be fixed to γ as well. The role of thumb is to set γ equal to a ratio of signal and noise variances, i.e., SNR $= 50$ dB $\Rightarrow \gamma = 10^5$ or SNR $= 20$ dB $\Rightarrow \gamma = 10^2$, etc.[4] Then, we have found that choosing $\alpha = 10^{-1}\gamma$, $\beta = 10^4\gamma$, and $\delta = 10^3\gamma$ usually results in good convergence. For higher noise levels (smaller γ), we observed that $\delta = 10^2\gamma$ is better.

In our experiments, the number of iteration in the main loop and in the u-step and h-step typically did not exceed ten. In order to further decrease computational time, we tried to modify the algorithm in several ways. For example, we found it very effective to divide the algorithm into two stages. In the first stage, we select a small (typically 256×256) central region from input images and run the algorithm on this selection. In the second stage, we take the estimated PSFs from the first stage and apply one u-step on the whole image in order to obtain the final reconstructed image. The usable output of the first stage are thus PSFs and not the reconstructed central region. We observed that fixing γ to ten (even for the SNR above 10 dB) in the first stage and

[3] A matrix of such size, if stored in double precision, occupies approximately 78 MB of memory, which current computers can easily handle.

[4] We use a standard definition of the signal to noise ratio, SNR $= 10\log(s^2/\sigma^2)$, where s^2 and σ^2 are the signal and noise variances, respectively.



(a) (b) (c)

Fig. 9. Test data set. (a) Original image 256 × 256. (b) Two blurs 9 × 9. (c) Example of an input blurry pair with SNR = 50 dB.

setting other parameters according to formulas as shown above produces accurate PSFs in a more reliable way. This convergence boost can be explained by noting that the reconstructed image for lower γ becomes more piecewise constant (patchy) with only strong edges preserved, which makes the h-step in the fidelity term focus only on areas around strong edges and neglect areas with details that are prone to noise.

Another modification, which proved to be a minor improvement, was to estimate PSFs in a multiscale fashion. Initializing with upsampled PSFs from the courser levels tend to decrease the number of iterations. However, we observed that more than two levels (half-sized and original scale) are not necessary and that the choice of the upsampling algorithm is important. Simple linear upsampling generates PSFs that are wider than the true PSFs on that scale, and we waste several iterations of the algorithm to shrink the PSFs back. In our tests, we were using a Lanczos interpolation method, which seems to give the best results.

To provide the cost of individual steps in terms of computer time, we performed blind deconvolution of two one-megapixel images with PSF size 40 × 40 on a 2.7-GHz Pentium Dual-Core CPU using our MATLAB implementation. The cost of one iteration inside the u-step and the h-step is around 0.8 and 4.5 s, respectively. Calculating matrix $\mathbf{R}$ using the whole images takes 11 min in this case, which is clearly the most time-consuming step. However, as pointed out earlier, we can calculate $\mathbf{R}$ on a small region. For example, for a 256 × 256 block, the calculation (same PSF size 40 × 40) then takes around 30 s.

VI. Experiments

In order to illustrate the favorable convergence properties of the proposed algorithm, we performed two sets of experiments. The first set works with synthetically blurred data and compares convergence and quality of PSF and image estimations for different SNRs and blur sizes. The second set of experiments compares the proposed algorithm with another MC blind deconvolution method of Katkovnik *et al.* [33] and demonstrates deconvolution of real photos taken with a standard digital camera.

The setup for the synthetic data experiment was the following. We took the Lena image in Fig. 9(a) and convolve it with two 9 × 9 blurs [see Fig. 9(b)] and add noise at three different levels SNR $= 50, 30,$ and10 dB. An example of blurry images for the least noisy case is in Fig. 9(c). To evaluate performance in every iteration j of the main loop, we use normalized root mean square error defined as NRMSE $= \|\widehat{\mathbf{h}}^j - \mathbf{h}^*\|/\|\mathbf{h}^*\|$, where $\widehat{\mathbf{h}}^j$ is the estimation of PSFs after j iterations and $\mathbf{h}^*$ are the true PSFs. NRMSE as a function of iterations and estimated PSFs for different situations are summarized in Fig. 10. NRMSE is plotted in logarithmic scale. Three graphs correspond to three levels of SNRs. In each case, we ran the algorithm with three different PSF supports: 9 × 9 (solid line), 15 × 15 (dotted line), and 21 × 21 (dashed line). The corresponding estimated sharp images for the PSF support 21 × 21 and are summarized in Fig. 11. One can see that the proposed method provides accurate results regardless of the degree of PSF size overestimation and shows robustness with respect to noise.

There are several interesting points we can draw from the obtained results. First of all, the MSE decreases very quickly. In most of the cases, after five iterations, MSE remains almost constant. For overestimated blur supports (dotted and dashed line) MSE reaches almost the same level as for the correct blur support (solid line), but the decrease is slightly less sharp (particularly visible for SNR = 50 dB). This is logical since, in the overestimated case, the dimensionality of the problem is higher, and the MC constraint $\mathbf{R}_\Delta$ is less effective, as discussed in Section IV-C. Clearly, as the noise level increases, the lowest attainable MSE increases as well. For SNR = 50 dB [see Fig. 10(a)], estimated PSFs are very accurate. The corresponding estimated image in Fig. 11(a) is almost perfect. For SNR = 30 dB, [see Fig. 10(b)], the estimated PSFs take the shape of the true PSFs but are slightly blurred. The estimated image in Fig. 11(b) still looks very sharp and artifact free. As the noise level increases further to SNR = 10 dB [see Fig. 10(c)], the quality of deconvolution starts to deteriorate, but the TV denoising feature of the method is evident, as shown in Fig. 11(c).

There are few data in the literature to which we can directly compare which uses multiple frames in the process. Most of the MC work presented in the introduction is mainly theoretical and presents no algorithms for large-scale problems. Comparison with single-channel results is possible, but we do not feel that this is fair to these other methods. To our knowledge, the only recent method, which is intrinsically MC and claims to work with large kernels, was proposed in [33]. This method performs alternating minimization by switching between minimization with respect to the image (corresponds to our u-step) and minimization with respect to the kernels (corresponds to our h-step). A variation of the steepest descent algorithm is used

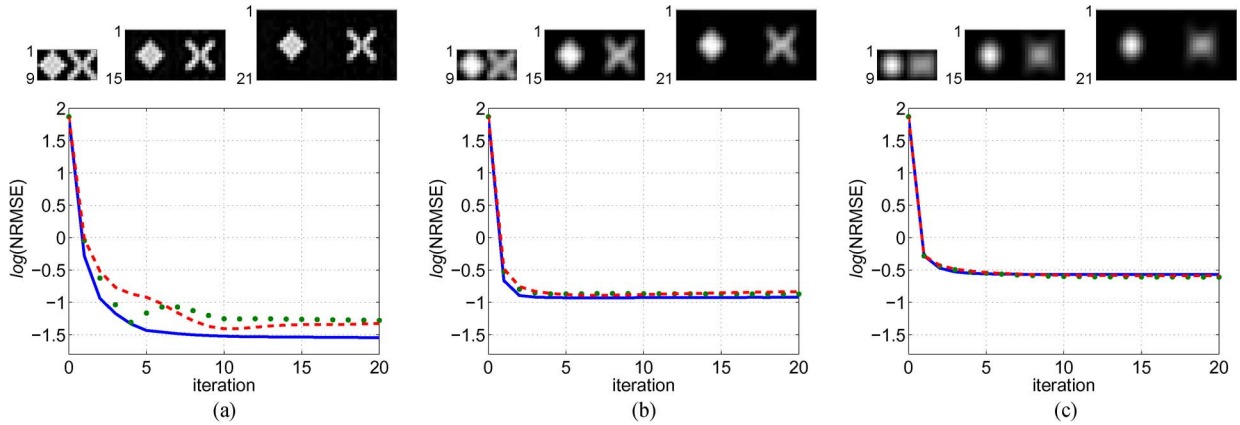


Fig. 10. Estimated PSFs and plots of NRMSE for different noise levels in input blurry images: (a) 50, (b) 30, and (c) 10 dB. Three different PSF supports were considered in each noisy case: (solid line) correct PSF size 9 × 9, (dotted line) two overestimated sizes 15 × 15, and (dashed line) 21 × 21. (a) 50 dB. (b) 30 dB. (c) 10 dB.

Fig. 11. Estimated sharp images for the PSF size set to 21 × 21 and three different noise levels: (a) 50, (b) 30, and (c) 10 dB. Results are arranged as in Fig. 10. The first row shows one of the input images and the second row shows the estimated image. (a) 50 dB. (b) 30 dB. (c) 10 dB.

for minimization. Everything is implemented in the Fourier domain, as in our case. For minimization, we use ALM in order to work with nonlinear regularization terms in an efficient manner. Katkovnik *et al.* use a variation of the steepest descent algorithm with only quadratic terms. Instead of using regularization, they project current estimation after every iteration into an admissible set of solutions (such as positive PSFs with limited support and image intensity values between 0 and 1) and perform spatially adaptive image denoising based on the intersection-of-confidence-interval rule. To compare the methods, we took a data set generated in [15], which contained four images blurred by eight PSFs providing 32 blurred images [see Fig. 12(a) and (b)]. The blurred images are real and captured by a digital camera. The ground-truth PSFs in Fig. 12(b) were estimated by a collection of point sources installed in the observed scene. We divided the blurred images into eight groups (each containing one image blurred by four blurs) and applied both methods. The NRMSE of the estimated images and blurs are plotted in Fig. 12(c) and (d). One can see that, in half of the cases, our method provides better PSFs (in the NRMSE sense) and outperforms the other method in the image NRMSE in all eight cases. In addition, our method requires only ten iterations of alternating minimization, whereas the other method requires roughly 100 iterations to achieve these results.[5]

In order to demonstrate that the algorithm works well in many practical applications, we took several pairs of images with a 3-megapixel digital camera Olympus C3020Z and applied the proposed algorithm. Light conditions were low, and the shutter speed of the camera was typically longer than 1/10 s. Such setting produces nice blurry images, when the camera is held in hands. It is of course necessary to first register the input photos before the algorithm can be applied. In our case, we do not have

[5]It is true that we perform at most ten iterations inside both u-step and h-step. Katkovnik's method cannot perform many iterations inside their u-step and h-step since they need to project into the admissible set frequently; therefore, they do ten steps of steepest descent in the h-step and one step in the u-step.

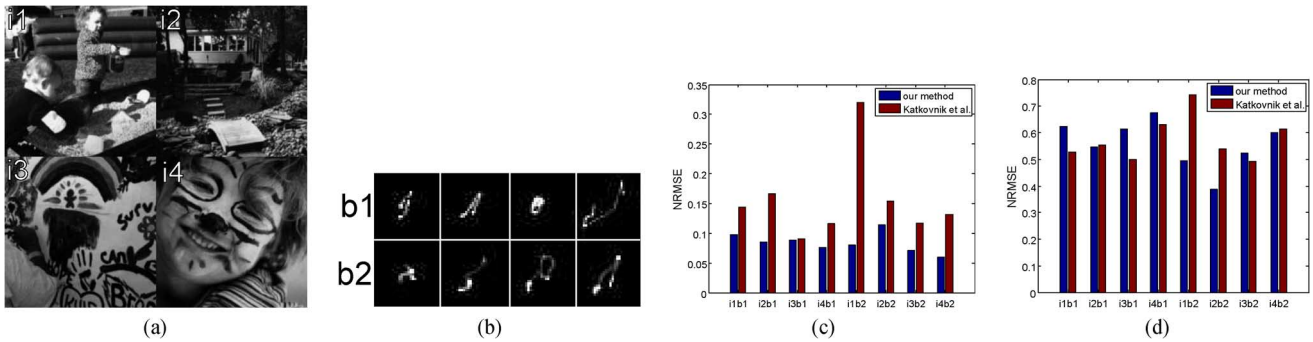


Fig. 12. Comparison with Katkovnik *et al* [33] Ground-truth data from Levin's data set [15]: (a) four images and (b) eight blur kernels, which generates 32 blurred images. We split the kernels into two groups (b1, b2) and got eight input sets each containing four blurred images. (c) NRMSE of estimated sharp images. (d) NRMSE of estimated kernels. Left bars are results of our method and right bars are results of [33].

Fig. 13. Real data set. (a) and (b) Ttwo input blurry images of size 2048 × 1536. (c) Estimated output sharp image using the proposed algorithm. (d) Closeups of the input images and the output, and estimated PSFs of size 50 × 30.

to deal with heavily misregistered data since the images have been taken one after another with minimum delay. A fast registration method, which proved to be adequate and was used in these experiments, works as follows. A reference image is selected from the input set $\{\mathbf{g}_k\}$, and the other images (called sensed images) are sequentially registered to the reference one. The reference and sensed image are first divided into several nonoverlapping blocks (typically 6×6). Phase correlation is applied in each block to determine the integer translation vector between the reference and the sensed block. The estimated shifts $(6 * 6 = 36)$ are used to calculate parameters of an affine transform. The sensed images are then interpolated using the estimated affine transforms.

Reconstruction results for two different data sets are in Figs. 13 and 14. Input image pairs exhibit relatively large blurring, but the reconstructed images are sharp and with negligible artifacts (see image closeups for better visual comparison). Estimated PSF pairs model very well motion blurs induced by camera shake. Some artifacts are visible in the second data set [see Fig. 14(c)] around the snow heap in the left bottom corner. It is very likely, that the blur is slightly different in this part due to a different distance from the camera or due to rotational movement during acquisition. Since our method assumes space-invariant blurs, such artifact are however inevitable.

VII. Conclusion

We have presented a new algorithm for solving MC blind deconvolution. The proposed approach starts by defining an optimization problem with image and blur regularization terms. To force sparse image gradients, the image regularizer is formu-

Fig. 14. Real data set. (a) and (b) Two input blurry images of size 2048 × 1536. (c) Estimated output sharp image using the proposed algorithm. (d) Closeups of the input images and the output and estimated PSFs of size 40 × 40.

lated using a standard isotropic TV. The PSF regularizer consists of two terms: MC constraint (matrix $\mathbf{R}_\Delta$) and sparsity–positivity. The MC constraint is improved by considering image Laplacian, which brings better noise robustness at little cost. Positivity helps the method to convergence to a correct solution, when the used PSF size is much larger than the true one. The proposed approach solves the optimization problem in an iterative way by alternating between minimization with respect to the image (u-step) and with respect to the PSFs (h-step). Sparsity and positivity imply nonlinearity, but by using the variable splitting and ALM (or split-Bregman method), we can solve each step efficiently, and moreover, convergence of each step is guaranteed. Experiments on synthetic data illustrate fast convergence of the algorithm, robustness to noise, and stability in the case of overestimated PSF sizes. Experiments on large real data underline practical aspects of the algorithm. Current and future work involves extending this approach to the space-variant blur and analyzing the convergence properties.

REFERENCES

[1] A. Levin, R. Fergus, F. Durand, and W. T. Freeman, "Image and depth from a conventional camera with a coded aperture," *ACM Trans. Graph.*, vol. 26, no. 3, p. 70, Jul. 2007.

[2] M. Elad, P. Milanfar, and R. Rubinstein, "Analysis versus synthesis in signal priors," *Inverse Probl.* vol. 23, no. 3, pp. 947–968, Jun. 2007 [Online]. Available: http://stacks.iop.org/0266-5611/23/i=3/a=007

[3] I. W. Selesnick and M. Figueiredo, "Signal restoration with overcomplete wavelet transforms: Comparison of analysis and synthesis priors," in *Proc. SPIE*, 2009, vol. 7446, p. 74 460D.

[4] T. Goldstein and S. Osher, "The split bregman method for l1-regularized problems," *SIAM J. Imag. Sci.* vol. 2, no. 2, pp. 323–343, Apr. 2009 [Online]. Available: http://portal.acm.org/citation.cfm?id=1658384.1658386

[5] M. V. Afonso, J. M. Bioucas-Dias, and M. A. T. Figueiredo, "Fast image recovery using variable splitting and constrained optimization," *IEEE Trans. Image Process.*, vol. 19, no. 9, pp. 2345–2356, Sep. 2010.

[6] G. Ayers and J. C. Dainty, "Iterative blind deconvolution method and its application," *Opt. Lett.*, vol. 13, no. 7, pp. 547–549, Jul. 1988.

[7] T. Chan and C. Wong, "Total variation blind deconvolution," *IEEE Trans. Image Process.*, vol. 7, no. 3, pp. 370–375, Mar. 1998.

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

[9] *Blind Image Deconvolution, Theory and Application*, P. Campisi and K. Egiazarian, Eds. Boca Raton, FL: CRC Press, 2007.

[10] R. Fergus, B. Singh, A. Hertzmann, S. T. Roweis, and W. T. Freeman, "Removing camera shake from a single photograph," in *Proc. SIGGRAPH: ACM SIGGRAPH Papers*, New York, 2006, pp. 787–794.

[11] J. Miskin and D. J. MacKay, "Ensemble learning for blind image separation and deconvolution," in *Advances in Independent Component Analysis*, M. Girolani, Ed. New York: Springer-Verlag, 2000.

[12] J. Jia, "Single image motion deblurring using transparency," in *Proc. IEEE Conf. CVPR*, Jun. 17–22, 2007, pp. 1–8.

[13] N. Joshi, R. Szeliski, and D. J. Kriegman, "PSF estimation using sharp edge prediction," in *Proc. IEEE CVPR*, Jun. 23–28, 2008, pp. 1–8.

[14] Q. Shan, J. Jia, and A. Agarwala, "High-quality motion deblurring from a single image," in *Proc. SIGGRAPH: ACM SIGGRAPH*, New York, 2008, pp. 1–10.

[15] A. Levin, Y. Weiss, F. Durand, and W. Freeman, "Understanding and evaluating blind deconvolution algorithms," in *Proc. IEEE Conf. CVPR*, 2009, pp. 1964–1971.

[16] S. Cho and S. Lee, "Fast motion deblurring," *ACM Trans. Graph. (SIGGRAPH ASIA)*, vol. 28, no. 5, p. 145, Dec. 2009.

[17] L. Xu and J. Jia, "Two-phase kernel estimation for robust motion deblurring," in *Proc. 11th ECCV*, Berlin, Germany, 2010, pp. 157–170 [Online]. Available: http://portal.acm.org/citation.cfm?id=1886063.1886077

[18] B. Zitová and J. Flusser, "Image registration methods: A survey," *Image Vis. Comput.*, vol. 21, no. 11, pp. 977–1000, Oct. 2003.

[19] M. Sorel and J. Flusser, "Space-variant restoration of images degraded by camera motion blur," *IEEE Trans. Image Process.*, vol. 17, no. 2, pp. 105–116, Feb. 2008.

[20] T. Schulz, "Multiframe blind deconvolution of astronomical images," *J. Opt. Soc. Am. A*, vol. 10, no. 5, pp. 1064–1073, May 1993.

[21] G. Harikumar and Y. Bresler, "Perfect blind restoration of images blurred by multiple filters: Theory and efficient algorithms," *IEEE Trans. Image Process.*, vol. 8, no. 2, pp. 202–219, Feb. 1999.

[22] G. Giannakis and R. Heath, "Blind identification of multichannel FIR blurs and perfect image restoration," *IEEE Trans. Image Process.*, vol. 9, no. 11, pp. 1877–1896, Nov. 2000.
[23] H.-T. Pai and A. Bovik, "On eigenstructure-based direct multichannel blind image restoration," *IEEE Trans. Image Process.*, vol. 10, no. 10, pp. 1434–1446, Oct. 2001.
[24] S. Pillai and B. Liang, "Blind image deconvolution using a robust GCD approach," *IEEE Trans. Image Process.*, vol. 8, no. 2, pp. 295–301, Feb. 1999.
[25] M. Haindl and S. Šimberová, "Model-based restoration of short-exposure solar images," in *Frontiers in Artificial Intelligence and Applications*, L. Jain and R. Howlett, Eds. Amsterdam, The Netherlands: IOS Press, 2002, pp. 697–706.
[26] G. Panci, P. Campisi, S. Colonnese, and G. Scarano, "Multichannel blind image deconvolution using the bussgang algorithm: Spatial and multiresolution approaches," *IEEE Trans. Image Process.*, vol. 12, no. 11, pp. 1324–1337, Nov. 2003.
[27] F. Šroubek and J. Flusser, "Multichannel blind deconvolution of spatially misaligned images," *IEEE Trans. Image Process.*, vol. 14, no. 7, pp. 874–883, Jul. 2005.
[28] L. Rudin, S. Osher, and E. Fatemi, "Nonlinear total variation based noise removal algorithms," *Phys. D*, vol. 60, no. 1–4, pp. 259–268, Nov. 1992.
[29] J. B. Reade, *Calculus With Complex Numbers*. Boca Raton, FL: CRC Press, 2003, ch. ch. Fundamental theorem of algebra, pp. 75–81.
[30] J. Yang, W. Yin, Y. Zhang, and Y. Wang, "A fast algorithm for edge-preserving variational multichannel image restoration," *SIAM J. Imag. Sci.* vol. 2, no. 2, pp. 569–592, Apr. 2009 [Online]. Available: http://portal.acm.org/citation.cfm?id=1658384.1658394
[31] J. Eckstein and D. P. Bertsekas, "On the Douglas-Rachford splitting method and the proximal point algorithm for maximal monotone operators," *Math. Program.*, vol. 55, no. 3, pp. 293–318, Jun. 1992.
[32] J. Moreau, "Proximité et dualité dans un espace hilbertien," *Bulletin de la Société Mathématique de France*, vol. 93, pp. 273–299, 1965.
[33] V. Katkovnik, D. Paliy, K. Egiazarian, and J. Astola, "Frequency domain blind deconvolution in multiframe imaging using anisotropic spatially-adaptive denoising," in *Proc. 14th EUSIPCO*, Sep. 2006, pp. 1–5.

Filip Šroubek (M'08) received the M.S. degree in computer science from the Czech Technical University, Prague, Czech Republic, in 1998 and the Ph.D. degree in computer science from Charles University, Prague, in 2003.

From 2004 to 2006, he was on a postdoctoral position with the Instituto de Optica, CSIC, Madrid, Spain. In 2010 and 2011, he was the Fulbright Visiting Scholar with the University of California, Santa Cruz. He is currently with the Institute of Information Theory and Automation and partially also with the Institute of Photonics and Electronics, Academy of Sciences of the Czech Republic, Prague. He is the author of seven book chapters and over 80 journal and conference papers on image fusion, blind deconvolution, super-resolution, and related topics.

Peyman Milanfar (F'10) received the B.S. degree in electrical engineering and mathematics from the University of California, Berkeley, in 1988 and the M.S., E.E., and Ph.D. degrees in electrical engineering from the Massachusetts Institute of Technology, Cambridge, in 1990, 1992, and 1993, respectively.

Until 1999, he was a Senior Research Engineer with SRI International, Menlo Park, CA. He is currently a Professor of electrical engineering with the University of California, Santa Cruz. From 1998 to 2000, he was a Consulting Assistant Professor of computer science with Stanford University, Stanford, CA. In 2002, he was also a Visiting Associate Professor with Standford University. His research interests include statistical signal, image processing, and inverse problems.

Dr. Milanfar is a member of the Signal Processing Society Image, Video, and Multidimensional Signal Processing Technical Committee. From 1998 to 2001, he was an Associate Editor for the IEEE Signal Processing Letters and was an Associate Editor for the IEEE TRANSACTIONS ON IMAGE PROCESSING from 2005–2010. He is currently on the editorial board of the *SIAM Journal of Imaging Science* and *Image and Vision Computing*. He was the recipient of the US National Science Foundation CAREER award, and the best paper award from the IEEE Signal Processing Society in 2010.