This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2016.2516949, IEEE Access

Fast Weighted Total Variation Regularization Algorithm for Blur Identification and Image Restoration

Haiying Liu⋆,†, *Member, IEEE,* Jason Gu†, *Senior Member, IEEE,*
Max Q.-H. Meng‡, *Fellow, IEEE* Wu-Sheng Lu∗, *Life Fellow, IEEE,*
⋆ School of Electrical Engineering and Automation, Qilu University of Technology, Jinan, 250353, China
† Department of Electrical and Computer Engineering, Dalhousie University, NS, B3H 4R2, Canada
‡ Department of Electronic Engineering, The Chinese University of Hong Kong, Hong Kong, China
∗ Department of Electrical and Computer Engineering, University of Victoria, BC, V8P 3W6, Canada
Corresponding author: Jason Gu (jason.gu@dal.ca)

***Abstract*—Images obtained from complicated environments are often blurred by unknown kernels and corrupted by measurement noise. This paper presents a new total variation (TV) minimization based method for blindly deburring such images. Unlike the alternating optimization based algorithms, the proposed method adopts a joint estimation strategy that estimates the unknown blurring kernel and unknown image in an iterative manner where each iteration performs two separate image denoising subproblems that admit fast implementation. Experimental results on gray-scale test images as well as synthetic images and color images are presented to demonstrate that the new method is capable of blindly deblurring images with considerably improved performance.**

***Index Terms*—Blind Deconvolution, TV Minimization, Image Denoising, Image Deblurring**

I. Introduction

CLASSICAL image restoration algorithm is dedicated to estimating the true image assuming the blur is known. Several methods solving these problems are available in the literature [1]–[9], including those based on total variation minimization which have been found especially effective. By contrast, blind deconvolution as an image restoration task tackles a much more difficult problem where, in addition to the image being unknown, the blurring mechanism is also unknown. TV-based image restoration algorithms [10]–[33] have been examined in connection to this type of problem for the blind deconvolution. An approach that turns out to be particularly successful for blind deconvolution is proposed in [10] and improved in [11], [13], [15], [24], [27] which is commonly known as the alternating optimization technique. The key idea of these aforementioned alternating optimization techniques is fixing one variable while the other is being optimized. Besides the above TV-based image restoration algorithm, there are other recent non-blind deconvolution papers that are relevant and need to be mentioned here. The algorithm based on Block Matching 3-D (BM3D) frames decoupled the denoising and deblurring operations shows superiority with respect to the state of the art in the field [36]. Some researchers also developed an iterative graph-based framework for image restoration based on a new definition of the normalized graph Laplacianmethods [34], [35] and their effectiveness for different restoration problems is shown.

The primary goal of this paper is to develop a new method for blind deconvolution of gray-scale as well as color images. Unlike the alternating-optimization based algorithm (which reduces the problem at hand to alternately performing two standard deblurring subproblems with known blurring kernels), the new method adopts a *joint* estimation strategy to synchronously estimate both the unknown kernel and unknown image. The motivation behind this strategy is that the problem of interest has one objective function with *two* unknowns, therefore we develop a technique to optimize both unknowns together rather than fixing one and optimizing the other alternately. To deal with the technical challenges arising from this joint estimation approach, a new local framework was proposed and allowed us to carry out the joint estimation of the unknowns in an iterative manner, where each iteration performs two separate image denoising subproblems which can be carried out efficiently using the techniques [5], [9]. Simulation results are presented to demonstrate that the proposed algorithm is capable of blindly deblurring images with considerably improved performance in comparison with well-established algorithms in literatures.

II. Background and Related Work

A. Image Model

We follow [2] to consider the basic image model

$$\boldsymbol{\mathcal{A}}\boldsymbol{u} + \boldsymbol{w} = \boldsymbol{u}_0 \tag{1}$$

where $\boldsymbol{\mathcal{A}}$ represents an affine map standing for a blurring operator, $\boldsymbol{u}_0$ denotes an observed noisy image of size $n_1 \times n_2$, and $\boldsymbol{w}$ is normally distributed additive noise. Assuming noise $\boldsymbol{w}$ is Gaussian white with independent and identically distributed components of zero mean and variance σ^2, the restoration problem here is to estimate (recover) image $\boldsymbol{u}$ and the blur operator $\boldsymbol{\mathcal{A}}$ given the observation $\boldsymbol{u}_0$. We stress that because both the blurring operator $\boldsymbol{\mathcal{A}}$ and image $\boldsymbol{u}$ are unknown, this is

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2016.2516949, IEEE Access

a blind deconvolution problem which makes model (1) become nonlinear, hence technically more challenging relative to some deburring problems encountered in reference papers [1]–[9].

B. TV *Norm of Gray-Scale Images*

Let $\boldsymbol{u} \in R^{n_1 \times n_2}$ represent a digital gray-scale image, the isotropic total variation ($\|\boldsymbol{u}\|_{\text{TV}^{(\text{I})}}$) of $\boldsymbol{u}$ is defined by

$$\|\boldsymbol{u}\|_{\text{TV}^{(\text{I})}} = \sum_{i=1}^{n_1}\sum_{j=1}^{n_2} \|\boldsymbol{D}_{i,j}\boldsymbol{u}\|_2, \quad \boldsymbol{D}_{i,j}\boldsymbol{u} = \begin{bmatrix} \boldsymbol{D}_{i,j}^{(h)}\boldsymbol{u} \\ \boldsymbol{D}_{i,j}^{(v)}\boldsymbol{u} \end{bmatrix} \quad (2)$$

where

$$\boldsymbol{D}_{i,j}^{(h)}\boldsymbol{u} = \begin{cases} u_{i+1,j} - u_{i,j}, & i < n_1, \\ 0, & i = n_1 \end{cases}$$

$$\boldsymbol{D}_{i,j}^{(v)}\boldsymbol{u} = \begin{cases} u_{i,j+1} - u_{i,j}, & j < n_2, \\ 0, & j = n_1 \end{cases} \quad (3)$$

C. *Related Work*

For blurred images with known blurring operator $\boldsymbol{\mathcal{A}}$, the image restoration problem can be treated by solving the unconstrained convex problem [2], [5], [6], [9]x

$$\underset{\boldsymbol{u}}{\text{minimize}} \quad \frac{1}{2}\|\boldsymbol{\mathcal{A}}\boldsymbol{u} - \boldsymbol{u}_0\|_F^2 + \mu\|\boldsymbol{u}\|_{\text{TV}^{(\text{I})}} \quad (4)$$

where $\|\cdot\|_F$ denotes Frobenius norm of matrix and $\mu > 0$ is regularization parameter that balances the trade-off between removing noise or small details.

For blurred images with an unknown blurring mechanism where the blur operator $\boldsymbol{\mathcal{A}}$ in model (1) is replaced by a convolutional kernel $\boldsymbol{a}$, thus the model becomes

$$\boldsymbol{a} * \boldsymbol{u} + \boldsymbol{w} = \boldsymbol{u}_0 \quad (5)$$

The image blind deconvolution problem has been investigated by many researchers, of particular relevance to the method proposed here are the techniques developed in [13] where the blind deblurring problem is formulated as the optimization problem

$$\min_{\boldsymbol{u},\boldsymbol{a}} F(\boldsymbol{u},\boldsymbol{a}) = \frac{1}{2}\|\boldsymbol{a} * \boldsymbol{u} - \boldsymbol{u}_0\|_2^2 + \beta_1\|\boldsymbol{u}\|_{\text{BV}}^2 + \beta_2\|\boldsymbol{a}\|_{\text{BV}}^2 \quad (6)$$

where $\|\cdot\|_{\text{BV}}$ is the bounded variation TV for image $\boldsymbol{u}$ and blur kernel $\boldsymbol{a}$ and defined as $\|\boldsymbol{u}\|_{\text{BV}} \overset{\text{def}}{=} \left\{ \int_\Omega \boldsymbol{u}\text{div}\varphi \text{d}x; \varphi = (\varphi_1, \varphi_2, \cdots \varphi_N \epsilon C_0^1(\Omega)^N), \|\varphi\|_{L^\infty}(\Omega) \le 1 \right\}$ denotes boundary variational TV norm as defined in [13]. Problem (6) is then solved by an alternating minimization strategy which starts by fixing kernel $\boldsymbol{a}$ to an initial estimate while minimizing functional $F(\boldsymbol{u},\boldsymbol{a})$ with respect to $\boldsymbol{u}$, then image $\boldsymbol{u}$ is fixed until the optimization result of $F(\boldsymbol{u},\boldsymbol{a})$ is obtained with respect to $\boldsymbol{a}$. The alternating procedure continues until both $\boldsymbol{a}$ and $\boldsymbol{u}$ converge, the convergence property of this algorithm is examined in [15].

D. *Fast Gradient-Based Algorithms for Problem (4)*

To solve problem (4) with a known blurring operator $\boldsymbol{\mathcal{A}}$, there is an especially effective algorithm developed in [5], [9] based on a fast gradient projection (FGP) algorithm, where each iteration solves a denoising subproblem using FGP. In this algorithm, the objective function in (4) is expressed as $f(\boldsymbol{u})+g(\boldsymbol{u})$ where $f(\boldsymbol{u}) = \frac{1}{2}\|\boldsymbol{\mathcal{A}}\boldsymbol{u}-\boldsymbol{u}_0\|_F^2$ is convex and smooth while $g(\boldsymbol{u}) = \mu\|\boldsymbol{u}\|_{\text{TV}}$ is convex but nonsmooth. Let $\boldsymbol{u}_k$ be the image obtained in the kth iteration, the next iterative $\boldsymbol{u}_{k+1}$ is generated by minimizing the proximal objective function which is quadratic and convex:

$$Q_L(\boldsymbol{u},\boldsymbol{u}_k) = f(\boldsymbol{u}_k) + \langle(\boldsymbol{u},\boldsymbol{u}_k), \nabla f(\boldsymbol{u}_k)\rangle + \frac{L}{2}\|\boldsymbol{u}-\boldsymbol{u}_k\|_F^2 + g(\boldsymbol{u}) \quad (7)$$

subject to: $\boldsymbol{u} \in C = \{\boldsymbol{u} : b_l \le \boldsymbol{u}_{p,q} \le b_u\}$.
where b_l and b_u are lower and upper bounds to the image ($b_l = 0$ and $b_u = 255$ for 8-bit digital images), and L denotes the Lipschitz constant for $\nabla f(\boldsymbol{u})$, i.e., for any $\boldsymbol{u}$ and $\boldsymbol{v}$,

$$\|\boldsymbol{\mathcal{A}}^{\text{T}}\boldsymbol{\mathcal{A}}(\boldsymbol{u} - \boldsymbol{v})\|_F \le L\|\boldsymbol{u} - \boldsymbol{v}\|_F \quad (8)$$

Note that the first three terms on the right-hand side of (7) can be combined into a complete square term , hence the solution of the minimization problem, denoted by $\boldsymbol{u}_{k+1} = P_L(\boldsymbol{u}_k)$, can be expressed as

$$\begin{aligned} P_L(\boldsymbol{u}_k) =& \underset{\boldsymbol{u}\in C}{\text{argmin}} \left\{ \frac{L}{2}\|\boldsymbol{u} - (\boldsymbol{u}_k - \frac{1}{L} \bigtriangledown h(\boldsymbol{u}_k))\|_F^2 + \frac{\mu}{L}\|\boldsymbol{u}\|_{\text{TV}} \right\} \quad (9) \\ =& \underset{\boldsymbol{u}\in C}{\text{argmin}} \left\{ \frac{L}{2}\|\boldsymbol{u} - \boldsymbol{b}_k\|_F^2 + \frac{\mu}{L}\|\boldsymbol{u}\|_{\text{TV}} \right\} \end{aligned}$$

where

$$\boldsymbol{b}_k = \boldsymbol{u}_k - \frac{1}{L}\boldsymbol{\mathcal{A}}^{\text{T}}\boldsymbol{\mathcal{A}}(\boldsymbol{u}_k - \boldsymbol{u}_0) \quad (10)$$

For implementation purpose, the Lipschitz constant L in (8) needs to be calculated. For the present case of $f(\boldsymbol{u})$, (8) implies that

$$L = \max_{\|\boldsymbol{u}\|_F=1} \|\boldsymbol{\mathcal{A}}^{\text{T}}\boldsymbol{\mathcal{A}}\boldsymbol{u}\|_F \quad (11)$$

From (9), it is clear finding the next iterate $\boldsymbol{u}_k$ amounts to solving an image denoising problem where the "noisy" image is represented by $\boldsymbol{b}_k$ and the item $\boldsymbol{b}_k$ can be calculated using equation (10). In this way, the deblurring problem (4) can be solved by iteratively solving a sequence of denoising problem as specified in (9) and (10).

III. NEW ALGORITHM BASED ON JOINT ESTIMATION OF KERNEL AND IMAGE

A. *The Solution Method at A Glance*

In this section, a novel algorithm is presented based on the TV-based formulation under the assumption that both $\boldsymbol{a}$ and $\boldsymbol{u}$ are unknown in the image model (5)

$$\min_{\boldsymbol{u},\boldsymbol{a}} F(\boldsymbol{u},\boldsymbol{a}) = \frac{1}{2}\|\boldsymbol{a} * \boldsymbol{u} - \boldsymbol{u}_0\|_F^2 + \mu_1\|\boldsymbol{u}\|_{\text{TV}^{(\text{I})}} + \mu_2\|\boldsymbol{a}\|_{\text{TV}^{(\text{I})}} \quad (12)$$

This strategy represents a major departure from the well established methods using alternating minimization where one of the unknowns is temporarily fixed while the other is being optimized, and the procedure continues in an alternating manner until both the kernel and image estimations converge. The difficulty encountered in a joint estimation method is that allowing both $\boldsymbol{a}$ and $\boldsymbol{u}$ to be unknown variables will make (5) become nonlinear and destroy the convexity of function $F(\boldsymbol{u},\boldsymbol{a})$ in (12). To deal with the aforementioned difficulty, we propose to work with (12) locally. Let $\boldsymbol{u}_k$ and $\boldsymbol{a}_k$ be the image and kernel obtained from the kth iteration, and we seek to find new estimates of the kernel and image in the form of $\boldsymbol{u} = \boldsymbol{u}_k + \triangle\boldsymbol{u}$ and $\boldsymbol{a} = \boldsymbol{a}_k + \triangle\boldsymbol{a}$ where the increments $\triangle\boldsymbol{u}$ and $\triangle\boldsymbol{a}$ denote small increments of $\boldsymbol{u}_k$ and $\boldsymbol{a}_k$, respectively. Namely, $\triangle\boldsymbol{u}$ and $\triangle\boldsymbol{a}$ are matrices of the same size with $\boldsymbol{u}_k$ and $\boldsymbol{a}_k$, whose elements are *small* in magnitude. With such constrains on $\triangle\boldsymbol{u}$ and $\triangle\boldsymbol{a}$, $\boldsymbol{u} = \boldsymbol{u}_k + \triangle\boldsymbol{u}$ and $\boldsymbol{a} = \boldsymbol{a}_k + \triangle\boldsymbol{a}$ will vary in a small vicinity of $\boldsymbol{u}_k$ and $\boldsymbol{a}_k$, respectively. Under this assumption, we can write

$$\begin{aligned}\boldsymbol{a}*\boldsymbol{u} =&\boldsymbol{a}_k*\boldsymbol{u}_k+\boldsymbol{a}_k*\triangle\boldsymbol{u}+\triangle\boldsymbol{a}*\boldsymbol{u}_k+\triangle\boldsymbol{a}*\triangle\boldsymbol{u}\\ =&(\boldsymbol{a}_k*\boldsymbol{u}_k+\triangle\boldsymbol{a}*\boldsymbol{u}_k)+(\boldsymbol{a}_k*\triangle\boldsymbol{u}+\boldsymbol{a}_k*\boldsymbol{u}_k)\\ &-\boldsymbol{a}_k*\boldsymbol{u}_k+\triangle\boldsymbol{a}*\triangle\boldsymbol{u}\\ =&(\boldsymbol{a}_k+\triangle\boldsymbol{a})*\boldsymbol{u}_k+(\triangle\boldsymbol{u}+\boldsymbol{u}_k)*\boldsymbol{a}_k\\ &-\boldsymbol{a}_k*\boldsymbol{u}_k+\triangle\boldsymbol{a}*\triangle\boldsymbol{u}\\ =&\boldsymbol{a}*\boldsymbol{u}_k+\boldsymbol{u}*\boldsymbol{a}_k-\boldsymbol{a}_k*\boldsymbol{u}_k+\triangle\boldsymbol{a}*\triangle\boldsymbol{u}\\ \approx&\boldsymbol{a}*\boldsymbol{u}_k+\boldsymbol{a}_k*\boldsymbol{u}-\boldsymbol{a}_k*\boldsymbol{u}_k\end{aligned} \tag{13}$$

The above approximation for the last line of (13) is valid because both $\triangle\boldsymbol{u}$ and $\triangle\boldsymbol{a}$ are small. In this way, the problem (12) at hand is simplified to

$$\min_{\boldsymbol{u},\boldsymbol{a}} F(\boldsymbol{u},\boldsymbol{a})=\frac{1}{2}\|\boldsymbol{a}*\boldsymbol{u}_k+\boldsymbol{a}_k*\boldsymbol{u}-\boldsymbol{f}_k\|_F^2+\mu_1\|\boldsymbol{u}\|_{\mathrm{TV^{(I)}}}+\mu_2\|\boldsymbol{a}\|_{\mathrm{TV^{(I)}}} \tag{14}$$

where $\boldsymbol{f}_k = \boldsymbol{a}_k*\boldsymbol{u}_k+\boldsymbol{u}_0$. It is important to note that when both $\boldsymbol{u}$ and $\boldsymbol{a}$ are unknown, equation (14) still remains convex with respect to $\{\boldsymbol{u},\boldsymbol{a}\}$.

For notation simplicity, we will denote $\boldsymbol{x} = \{\boldsymbol{u},\boldsymbol{a}\}$ and define a linear operator $\boldsymbol{\mathcal{A}}_k$ by $\boldsymbol{\mathcal{A}}_k\boldsymbol{x} = \boldsymbol{a}*\boldsymbol{u}_k+\boldsymbol{a}_k*\boldsymbol{u}$. Also, we denote $\|\boldsymbol{x}\|_{\mathrm{TV},\mu} = \mu_1\|\boldsymbol{u}\|_{\mathrm{TV^{(I)}}}+\mu_2\|\boldsymbol{a}\|_{\mathrm{TV^{(I)}}}$, the problem (14) now becomes

$$\min_{\boldsymbol{x}} F(\boldsymbol{x}) = \frac{1}{2}\|\boldsymbol{\mathcal{A}}_k\boldsymbol{x}-\boldsymbol{f}_k\|_F^2+\|\boldsymbol{x}\|_{\mathrm{TV^{(I)}},\mu} \tag{15}$$

and the minimization of $F(\boldsymbol{x})$ in (15) is carried out subject to the following constrains:

$$\sum_{i=-m}^{m}\sum_{j=-m}^{m}\boldsymbol{a}_{i,j}=1 \tag{16a}$$

$$\boldsymbol{u}\geq 0, \boldsymbol{a}\geq 0 \tag{16b}$$

$$|\boldsymbol{u}_{p,q}^{(k+1)}-\boldsymbol{u}_{p,q}^{(k)}|\leq\beta_1\cdot I_{\boldsymbol{u}} \tag{16c}$$

$$|\boldsymbol{a}_{i,j}^{(k+1)}-\boldsymbol{a}_{i,j}^{(k)}|\leq\beta_2\cdot I_{\boldsymbol{a}} \tag{16d}$$

$$b_l\leq\boldsymbol{u}_{p,q}\leq b_u \tag{16e}$$

$$\boldsymbol{a}_{i,j}=\boldsymbol{a}_{-i,-j},\ \text{for } 0\leq i,j\leq m \tag{16f}$$

where $\beta_1 \geq 0$ and $\beta_2 \geq 0$ are two small constants, $I_{\boldsymbol{u}}$, $I_{\boldsymbol{a}}$ are two defined unit matrices with the same size image $\boldsymbol{u}$ and kernel $\boldsymbol{a}$. Constrains (16a) and (16f) impose normalization and symmetry conditions on the kernel; (16b) requires that both kernel and image are nonnegative; (16c) and (16d) ensure that the increments $\triangle\boldsymbol{u}$ and $\triangle\boldsymbol{a}$ are small in magnitude; and (16e) requires image $\boldsymbol{u}$ to be a bounded valued.

By replacing $F(\boldsymbol{x})$ in (15) with an approximate proximal objective function, the problem of minimizing $F(\boldsymbol{x})$ in (15) is equivalent to separately minimizing two much simplified objective functions based on the properties of the Frobnius norm as follows:

$$\min_{\boldsymbol{u}} F(\boldsymbol{u}) = \frac{1}{2}\|\boldsymbol{u}-\boldsymbol{b}_{\boldsymbol{u}_k}\|_F^2+\frac{\mu_1}{L}\|\boldsymbol{u}\|_{\mathrm{TV^{(I)}}} \tag{17}$$

and

$$\min_{\boldsymbol{a}} F(\boldsymbol{a}) = \frac{1}{2}\|\boldsymbol{a}-\boldsymbol{b}_{\boldsymbol{a}_k}\|_F^2+\frac{\mu_2}{L}\|\boldsymbol{a}\|_{\mathrm{TV^{(I)}}} \tag{18}$$

where $\boldsymbol{b}_{\boldsymbol{u}_k}$ and $\boldsymbol{b}_{\boldsymbol{a}_k}$ are known quantities in the kth iteration of the algorithm. The derivation details (leading to (17) and (18)) are given in Sec.III.*B*. Consequently, the blind deblurring problem at hand can be solved by iteratively performing two denoising subproblems:

$$\min_{\boldsymbol{u}} F(\boldsymbol{u}) = \frac{1}{2}\|\boldsymbol{u}-\boldsymbol{b}_{\boldsymbol{u}_k}\|_F^2+\frac{\mu_1}{L}\|\boldsymbol{u}\|_{\mathrm{TV^{(I)}}} \tag{19a}$$

subject to:

$$\boldsymbol{u}\geq 0 \tag{19b}$$

$$|\boldsymbol{u}_{p,q}^{(k+1)}-\boldsymbol{u}_{p,q}^{(k)}|\leq\beta_1\cdot I_{\boldsymbol{u}} \tag{19c}$$

$$b_l\leq\boldsymbol{u}_{p,q}\leq b_u \tag{19d}$$

where $1\leq p\leq n_1, 1\leq q\leq n_2$ and

$$\min_{\boldsymbol{a}} F(\boldsymbol{a}) = \frac{1}{2}\|\boldsymbol{a}-\boldsymbol{b}_{\boldsymbol{a}_k}\|_F^2+\frac{\mu_2}{L}\|\boldsymbol{a}\|_{\mathrm{TV^{(I)}}} \tag{20a}$$

subject to:

$$\sum_{i=-m}^{m}\sum_{j=-m}^{m}\boldsymbol{a}_{i,j}=1 \tag{20b}$$

$$\boldsymbol{a}\geq 0 \tag{20c}$$

$$|\boldsymbol{a}_{i,j}^{(k+1)}-\boldsymbol{a}_{i,j}^{(k)}|\leq\beta_2\cdot I_{\boldsymbol{a}} \tag{20d}$$

$$\boldsymbol{a}_{i,j}=\boldsymbol{a}_{-i,-j} \tag{20e}$$

where $0\leq i,j\leq m$.

It is important to stress that both problems (19) and (20) are essentially standard image denoising problems to which fast algorithms such as those proposed in [5], [9] are applicable.

B. Analytical Details of the Proposed Algorithm

Below we will provide analytical details of the three key issues involved in the proposed algorithm, namely symmetry property of the kernel, gradient of the first term of $F(\boldsymbol{x})$ in (15) and derivation of problems (17) and (18).

(i) Symmetry of Kernel $\boldsymbol{a}$

For Gaussian lowpass blur, average blur and circularly averaging blur, the kernel matrix $\boldsymbol{a}$ is quadrantally symmetric (Actually we assumed a specific symmetric structure for the kernel throughout the paper, but the proposed algorithms can absolutely extend to some other regular type of blur kernels), thus only its upper left part, denoted by $\widetilde{\boldsymbol{a}}$, should be considered as the designed variable because the rest of matrix $\boldsymbol{a}$ can be constructed using $\widetilde{\boldsymbol{a}}$ as

$$\boldsymbol{a}=\widetilde{\mathbf{I}}\,\widetilde{\boldsymbol{a}}\,\widetilde{\mathbf{I}}^{\mathrm{T}} \tag{21}$$

where $\boldsymbol{a}$ is assumed to be size $M\times M$ with $M=2m+1$.

$$\widetilde{\mathbf{I}}=\mathbf{I}_{m+1}\widehat{\mathbf{I}} \tag{22}$$

$\widehat{\mathbf{I}}=[\mathrm{fliplr}(\mathbf{I}_m)\quad \mathbf{0}]$, $\mathrm{fliplr}(\mathbf{I}_m)$ is obtained by flipping identity matrix $\mathbf{I}_m$ from left to right.

(ii) Calculate gradient of $h(\boldsymbol{x})=\frac{1}{2}\|\boldsymbol{\mathcal{A}}_k\boldsymbol{x}-\boldsymbol{f}_k\|_F^2$

By the definitions of $\boldsymbol{\mathcal{A}}_k$ and $\boldsymbol{x}$ (see Sec.III.*A*.), the adjoint of $\boldsymbol{\mathcal{A}}_k$, denoted by $\boldsymbol{\mathcal{A}}_k^{\mathrm{T}}$, can be found as follows:

$$\begin{aligned}\langle\boldsymbol{\mathcal{A}}_k\boldsymbol{x},\boldsymbol{v}\rangle &= \langle\boldsymbol{a}*\boldsymbol{u}_k+\boldsymbol{a}_k*\boldsymbol{u},\boldsymbol{v}\rangle\\ &=\langle\boldsymbol{a}*\boldsymbol{u}_k,\boldsymbol{v}\rangle+\langle\boldsymbol{a}_k*\boldsymbol{u},\boldsymbol{v}\rangle\\ &=\langle\boldsymbol{a},\hat{\boldsymbol{u}}_k*\boldsymbol{v}\rangle+\langle\boldsymbol{u},\hat{\boldsymbol{a}}_k*\boldsymbol{v}\rangle\\ &=\langle(\boldsymbol{u},\boldsymbol{a}),\{\hat{\boldsymbol{a}}_k*\boldsymbol{v},\hat{\boldsymbol{u}}_k*\boldsymbol{v}\}\rangle\\ &=\langle\boldsymbol{x},\boldsymbol{\mathcal{A}}_k^T\boldsymbol{v}\rangle\end{aligned} \tag{23}$$

hence $\boldsymbol{\mathcal{A}}_k^T\boldsymbol{v}=\{\hat{\boldsymbol{a}}_k*\boldsymbol{v},\hat{\boldsymbol{u}}_k*\boldsymbol{v}\}$, where $\hat{\boldsymbol{a}}_{i,j}^{(k)}=\boldsymbol{a}_{-i,-j}^{(k)}$, $-m\leq i,j\leq m$, $\hat{\boldsymbol{u}}_{p,q}^{(k)}=\boldsymbol{u}_{-p,-q}^{(k)}$, $-m\leq p\leq n_1$, $-m\leq q\leq n_2$, to obtain an expression of $\hat{\boldsymbol{u}}_k$, we compute

$$\begin{aligned}\langle\boldsymbol{a}*\boldsymbol{u}_k,\boldsymbol{v}\rangle &= \sum_{p=1}^{n_1}\sum_{q=1}^{n_2}\left(\sum_{i=-m}^{m}\sum_{j=-m}^{m}\boldsymbol{a}_{i,j}\boldsymbol{u}_{p-i,q-j}^{(k)}\right)\cdot\boldsymbol{v}_{p,q}\\ &=\sum_{i=-m}^{m}\sum_{j=-m}^{m}\boldsymbol{a}_{i,j}\left(\sum_{p=1}^{n_1}\sum_{q=1}^{n_2}\boldsymbol{u}_{p-i,q-j}^{(k)}\cdot\boldsymbol{v}_{p,q}\right)\\ &=\langle\boldsymbol{a},\hat{\boldsymbol{u}}_k*\boldsymbol{v}\rangle\end{aligned} \tag{24}$$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2016.2516949, IEEE Access

Thus $\hat{\boldsymbol{u}}_k * \boldsymbol{v} = \sum_{p=1}^{n_1}\sum_{q=1}^{n_2} \boldsymbol{u}^{(k)}_{p-i,q-j} \cdot \boldsymbol{v}_{p,q}$ which implies $\hat{\boldsymbol{u}}^{(k)}_{p,q} = \boldsymbol{u}^{(k)}_{-p,-q}$.

To obtain an expression of $\hat{\boldsymbol{a}}_k$, here we assume periodic boundary extension for image and then compute:

$$\begin{aligned}
&\langle \boldsymbol{a}_k * \boldsymbol{u}, \boldsymbol{v}\rangle \\
&= \sum_{p=1}^{n_1}\sum_{q=1}^{n_2}\Big(\sum_{i=-m}^{m}\sum_{j=-m}^{m} \boldsymbol{a}^{(k)}_{i,j}\boldsymbol{u}_{p-i,q-j}\Big)\cdot \boldsymbol{v}_{p,q} \\
&= \sum_{i=-m}^{m}\sum_{j=-m}^{m} \boldsymbol{a}^{(k)}_{-i,-j}\Big(\sum_{p=1}^{n_1}\sum_{q=1}^{n_2} \boldsymbol{u}_{p,q}\cdot \boldsymbol{v}_{p-i,q-j}\Big) \\
&\text{let} \quad p' = p-i, q' = q-j, \\
&= \sum_{i=-m}^{m}\sum_{j=-m}^{m} \boldsymbol{a}^{(k)}_{i,j}\Big(\sum_{p'=1-i}^{n_1-i}\sum_{q'=1-j}^{n_2-j} \boldsymbol{u}_{p',q'}\cdot \boldsymbol{v}_{p'+i,q'+j}\Big) \qquad (25)\\
&\text{let} \quad \boldsymbol{a}^{(k)}_{-i',-j'} \triangleq \hat{\boldsymbol{a}}^{(k)}_{i,j}, i'=i, j'=j, \text{ then} \\
&= \sum_{i=-m}^{m}\sum_{j=-m}^{m} \hat{\boldsymbol{a}}^{(k)}_{i,j}\Big(\sum_{p=1+i}^{n_1+i}\sum_{q=1+j}^{n_2+j} \boldsymbol{u}_{p,q}\cdot \boldsymbol{v}_{p-i,q-j}\Big) \\
&= \sum_{p=1}^{n_1}\sum_{q=1}^{n_2} \boldsymbol{u}_{p,q}\Big(\sum_{i=-m}^{m}\sum_{j=-m}^{m} \boldsymbol{a}^{(k)}_{-i,-j}\cdot \boldsymbol{v}_{p-i,q-j}\Big) \\
&= \langle \boldsymbol{u}, \hat{\boldsymbol{a}}_k * \boldsymbol{v}\rangle
\end{aligned}$$

which implies $\hat{\boldsymbol{a}}^{(k)}_{i,j} = \boldsymbol{a}^{(k)}_{-i,-j}$.

Using the equations of (21), (22) and (23), the gradient of $h(\boldsymbol{x})$ can be computed as the following formula

$$\bigtriangledown h(\boldsymbol{x}) = \{\hat{\boldsymbol{a}}_k * (\boldsymbol{\mathcal{A}}_k\boldsymbol{x} - \boldsymbol{f}_k), \widetilde{\mathbf{I}}^{\mathrm{T}}\,[\hat{\boldsymbol{u}}_k * (\boldsymbol{\mathcal{A}}_k\boldsymbol{x} - \boldsymbol{f}_k)]\,\widetilde{\mathbf{I}}\} \qquad (26)$$

where variable $\boldsymbol{x}$ is redefined as $\boldsymbol{x} = \{\boldsymbol{u}, \tilde{\boldsymbol{a}}\}$.

(iii) Derivation of (17) and (18)

We start with considering the minimization of the proximal objective function of $F(\boldsymbol{x})$ in (15):

$$Q(\boldsymbol{x}, \boldsymbol{x}_k) = h(\boldsymbol{x}_k) + \langle \boldsymbol{x} - \boldsymbol{x}_k, \bigtriangledown h(\boldsymbol{x}_k)\rangle + \frac{L}{2}\|\boldsymbol{x} - \boldsymbol{x}_k\|_F^2 + \|\boldsymbol{x}\|_{\mu,\mathrm{TV}} \quad (27)$$

where L is a Lipschitz constant of $\bigtriangledown h(\boldsymbol{x})$. It follows that up to a constant the first three terms on the right-hand side of (27) can be combined into a complete square term, i.e.,

$$Q(\boldsymbol{x}, \boldsymbol{x}_k) = \frac{L}{2}\|\boldsymbol{x} - (\boldsymbol{x}_k - \frac{1}{L}\bigtriangledown h(\boldsymbol{x}_k))\|_F^2 + \|\boldsymbol{x}\|_{\mu,\mathrm{TV}^{(\mathrm{I})}} \qquad (28)$$

If we let

$$\boldsymbol{b}_k = \boldsymbol{x}_k - \frac{1}{L}\bigtriangledown h(\boldsymbol{x}_k) \triangleq (\boldsymbol{b}_{\tilde{\boldsymbol{a}}_k}, \boldsymbol{b}_{\boldsymbol{u}_k}) \qquad (29)$$

then (27) can be written as

$$\begin{aligned}
Q(\boldsymbol{x}, \boldsymbol{x}_k) = &\frac{1}{2}\|\boldsymbol{u} - \boldsymbol{b}_{\boldsymbol{u}_k}\|_F^2 + \frac{1}{2}\|\tilde{\boldsymbol{a}} - \boldsymbol{b}_{\tilde{\boldsymbol{a}}_k}\|_F^2 \\
&+ \frac{\mu_1}{L}\|\boldsymbol{u}\|_{\mu,\mathrm{TV}} + \frac{\mu_2}{L}\|\tilde{\boldsymbol{a}}\|_{\mu,\mathrm{TV}}
\end{aligned} \qquad (30)$$

Note that the first and third terms of (30) only depend on variable $\boldsymbol{u}$, while the other two terms only depend on variable $\tilde{\boldsymbol{a}}$. According to the properties of the Frobenius norm, the minimization of $Q(\boldsymbol{x}, \boldsymbol{x}_k)$ can be accomplished by two *separate* minimizations, namely

$$\min_{\boldsymbol{u}} \frac{1}{2}\|\boldsymbol{u} - \boldsymbol{b}_{\boldsymbol{u}_k}\|_F^2 + \frac{\mu_1}{L}\|\boldsymbol{u}\|_{\mathrm{TV}^{(\mathrm{I})}} \qquad (31)$$

which is identical to (17), and

$$\min_{\tilde{\boldsymbol{a}}} \frac{1}{2}\|\tilde{\boldsymbol{a}} - \boldsymbol{b}_{\tilde{\boldsymbol{a}}_k}\|_F^2 + \frac{\mu_2}{L}\|\tilde{\boldsymbol{a}}\|_{\mathrm{TV}^{(\mathrm{I})}} \qquad (32)$$

which is a simplified version of (18) with $\tilde{\boldsymbol{a}}$ replacing $\boldsymbol{a}$ in the light of symmetry property of $\boldsymbol{a}$, as discussed earlier.

Based on the analysis and derivation what has been done in Sec III.*A* and Sec III.*B*, the proposed algorithm for blind deconvolution can be outlined as **Algorithm 1**.

Algorithm 1 : Implementation of the proposed algorithm.

Input:

Input image $\boldsymbol{u}_0$, parameters $m, L, \delta, \mu_1, \mu_2, b_l, b_u, \beta_1$ and β_2, and number of iterations K; set iteration counter $k = 0$, initial $\boldsymbol{u}$ to be $\boldsymbol{u}_0$, and initial $\tilde{\boldsymbol{a}}$ to be the 2nd quadrant submatrix of the δ-matrix of size $(2m+1) \times (2m+1)$.

if $k < K$ **then**

- Use (26) and (29) to compute and update $\boldsymbol{b}_{\tilde{\boldsymbol{a}}_k}$ and $\boldsymbol{b}_{\boldsymbol{u}_k}$.
- Solve problems (31) and (32) to get $\boldsymbol{u}_k$ and $\tilde{\boldsymbol{a}}_k$, respectively.
- Set $k = k+1$. If $k = K$, output solution $\{\boldsymbol{u}_k, \boldsymbol{a}_k\}$ and terminate, otherwise repeat the loop until the algorithm iteration proceed to convergence.

end if

Output:

Optimal solution $\{\boldsymbol{u}_k, \boldsymbol{a}_k\}$.

C. Blind Deblurring of Color Images

A color image with the size of $n_1 \times n_2$ is represented by $\boldsymbol{u} = \{\boldsymbol{u}^{(1)}, \boldsymbol{u}^{(2)}, \boldsymbol{u}^{(3)}\}$, where $\boldsymbol{u}^{(1)}, \boldsymbol{u}^{(2)}$ and $\boldsymbol{u}^{(3)}$ are the components of $\boldsymbol{u}$ in red (R), green (G) and blue (B) channels, respectively. For color images, the model in (5) still remains valid where $\boldsymbol{a} = \{\boldsymbol{a}^{(1)}, \boldsymbol{a}^{(2)}, \boldsymbol{a}^{(3)}\}$ and $\boldsymbol{w} = \{\boldsymbol{w}^{(1)}, \boldsymbol{w}^{(2)}, \boldsymbol{w}^{(3)}\}$, so model (5) is equivalent to three single-channel models

$$\boldsymbol{a}^{(i)} * \boldsymbol{u}^{(i)} + \boldsymbol{w}^{(i)} = \boldsymbol{u}_0^{(i)} \text{ for } i = 1, 2, 3. \qquad (33)$$

Blind deblurring a color image is a problem of recovering image from observation $\boldsymbol{u}_0 = \{\boldsymbol{u}_0^{(1)}, \boldsymbol{u}_0^{(2)}, \boldsymbol{u}_0^{(3)}\}$, where the blurring kernel $\boldsymbol{a}$ is unknown. Based on model (30), blind deblurring a color image can be accomplished by solving three convex problems

$$\begin{aligned}
\min_{\boldsymbol{u}^{(i)}, \boldsymbol{a}^{(i)}} F_i(\boldsymbol{u}^{(i)}, \boldsymbol{a}^{(i)}) = &\frac{1}{2}\|\boldsymbol{a}^{(i)} * \boldsymbol{u}^{(i)} - \boldsymbol{u}_0^{(i)}\|_F^2 + \mu_1^{(i)}\|\boldsymbol{u}^{(i)}\|_{\mathrm{TV}^{(\mathrm{I})}} \\
&+ \mu_2^{(i)}\|\boldsymbol{a}^{(i)}\|_{\mathrm{TV}^{(\mathrm{I})}}
\end{aligned} \qquad (34)$$

for $i = 1, 2, 3$. On comparing (34) with (12), it's quite clear that each problems in (34) is identical to a problem of blind deconvolution of an image, hence it can be solved by applying **Algorithm 1**.

IV. PERFORMANCE EVALUATION

In this section, simulations are presented to illustrate the performance for our algorithm. The performance evaluation was carried out visually (i.e. subjectively) as well as numerically (i.e. objectively) from the improvement in peak-signal-to-noise ratio (PSNR) to structural similarity index measurement (SSIM). Besides these two evaluation standards, the elapsed time was also considered. All simulations were performed on a Windows7 laptop PC with an Intel(R) Core(TM) i3-4010U Duo CPU P8700@1.70 GHz with 4.0 GB of RAM.

Performance of our algorithm was compared with the alternating optimization based method first developed in the late 1990's and advanced in the early 2000's [10], [11], [13]–[15] and the latest algorithm which is proposed in [14]. To the authors' best knowledge, algorithm [14] remains to be one of the best (if not the best) methods available for blind deconvolution of still images. As mentioned earlier, state of the art deblurring algorithms such as those

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2016.2516949, IEEE Access

in [5] and [9] have been proposed, however the methods in [5], [9] assume known burring kernels. In our simulations, algorithm [13] reduces the blind deconvolution problem into two standard (non-blind) deconvolution subproblems, and was carried out using the fast convex programming solver provided by [9] for efficient and accurate solution. In other words, our algorithm was compared with an algorithm that combines the best of the methods of [13] and [9]. For this reason, we shall call it " [13]+ [9]" method in the rest of this section. The algorithm [33] solved blind deconvolution problem which the TV norm was replaced with it's quadratic upper bound. The approximation cuts off the high frequency information and the alternate minimization step resulted in quality degradation and computation, but the performance of algorithm [14] is better than the aforementioned algorithms; it's calculation of PSNR needs to extracted some pixels from the edge of the images, we just made comparison for gray-scale image *Lena* because the uncertainty result.

A. Blind deblurring of standard gray-scale images

For visual examination, the proposed algorithm was applied to standard classic image *Lena*, which was degraded by original 9×9 Gaussian kernel with standard deviation δ, which is 4 and additive Gaussian white noise with standard deviation $\sigma = 10^{-3}$. The original, blurred, and deblurred images of *Lena* by the [13] + [9], [14] and the proposed methods are shown in Fig.1 (a), (b), (c), (d) and (e). The numerical values of bounds β_1, β_2 and regularization parameters μ_1, μ_2 were set to $(\beta_1, \beta_2, \mu_1, \mu_2) = (2 \times 10^{-2}, 5 \times 10^{-3}, 6 \times 10^{-5}, 3 \times 10^{-5})$. It is observed that the PSNR has been improved from 23.4520 dB to 27.3840 dB, 28.1110 dB and **29.5081** dB via [13] + [9], [14] and the proposed method, respectively. The SSIM is improved from 0.6417 to 0.7656, 0.7654 and **0.8287**. The approximate eclipse time for algorithm [13] + [9] is 51.8859s, [14] is 539.4359s and our algorithm is 38.3606s. The original and identified kernels of Gaussian are depicted in Fig.2.

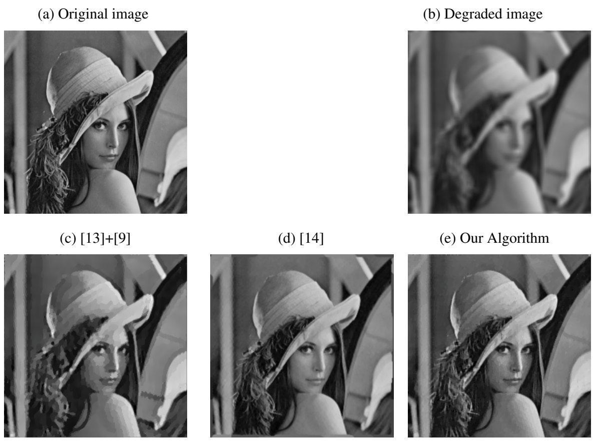

Fig. 1: (a) Original image, (b) blurred image, (c) deblurred by [13] + [9] method ($\lambda_1 = 1 \times 10^{-4}, \lambda_2 = 2 \times 10^{-4}$), (d) deblurred by [14] and (e) deblurred by the proposed algorithm.

The proposed algorithm is extended to standard gray images *Satellite* and *Cameraman* (256×256), *Boat* (512×512) with different kernels, where 9×9 Gaussian, 7×7 average lowpass and 9×9 circularly lowpass (also known as disk) blurs and additive Gaussian white noise with standard deviation $\sigma = 10^{-4}, 10^{-3}$ and 10^{-4} are imposed, respectively. The parameters were set as $(\beta_1, \beta_2, \mu_1, \mu_2) = (0.06, 0.002, 1 \times 10^{-4}, 2 \times 10^{-5})$ for *Satellite* and *Boat*, and $(0.05, 0.004, 1 \times 10^{-4}, 2 \times 10^{-5})$ for *Cameraman*, respectively. The compared results for *Satellite* are depicted in Fig.3, the compared SSIM, PSNR, identified kernel L2-error and the eclipse time between our and [13] + [9] methods are listed in TABLE I and the other images compared results are shown in TABLE II.

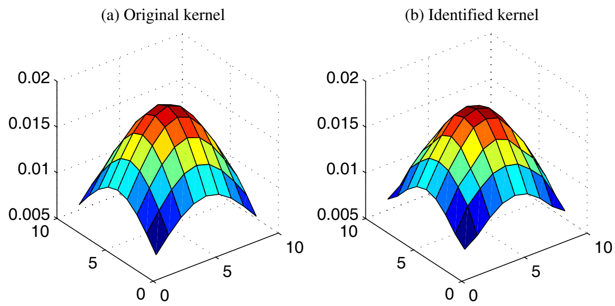

Fig. 2: (a) Original and (b) the identified Gaussian kernel using the proposed algorithm with the L2-error of 0.0008.

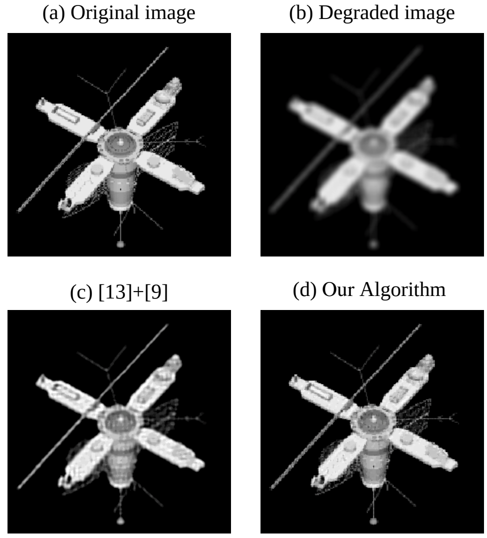

Fig. 3: (a) Original image, (b) blurred image, (c) deblurred by [13] + [9] method ($\lambda_1 = 3 \times 10^{-4}, \lambda_2 = 1 \times 10^{-5}$), (d) output of our algorithm.

TABLE I: Performance comparison for gray image *Satellite*.

Method / *Satellite*	[13] + [9]	Proposed Algorithm
SSIM	0.7595→0.9125	**0.7595→0.9739**
PSNR (dB)	21.4131→25.6472	**21.4131→31.4932**
Identified kernel L2 Error	0.0090	**0.0033**
Eclipse Time (s)	77.9242	**48.9063**

TABLE II: Performance comparison of standard gray images.

PSNR / Blur(Images)	PSNR before	PSNR after	
		[13] + [9]	Proposed Algorithm
Gaussian (*Satellite*)	21.4131	25.6472	**31.3250**
Average (*Boat*)	24.9657	28.0963	**32.0029**
Disk (*Cameraman*)	21.6917	22.8786	**29.0434**
E(t) (*Satellite*) (s)	\	77.9242	**48.9063**
E(t) (*Boat*) (s)	\	217.2793	**127.9388**
E(t) (*Cameraman*) (s)	\	16.4455	**3.4942**

B. Blind deblurring of natural gray-scale images

The natural images *Lifebuoy*, *Axe*, *Fence*, *Bird*, *Flag*, *Building* as depicted in Fig.4 were used to validate the efficiency of the proposed algorithm. The improvement of PSNR, SSIM and the regulated parameters for them are available in TABLE III, respectively as

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2016.2516949, IEEE Access

compared with [13] + [9] method which is the classic algorithms combination.

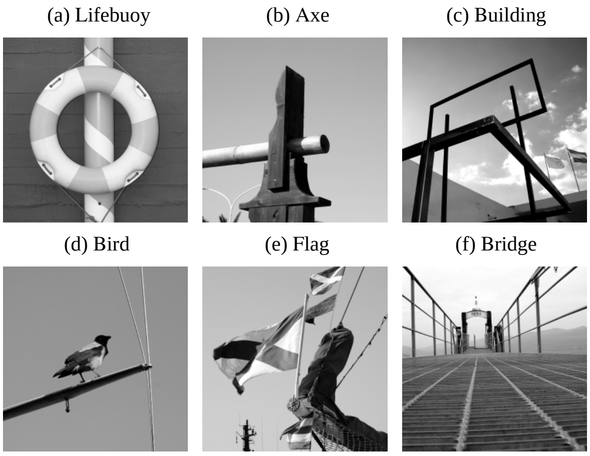

Fig. 4: Different natural gray-scale images.

Lifebuoy (240×240) is degraded by 9×9 Gaussian blurs and additive Gaussian white noise with standard deviation $\sigma = 10^{-4}$. The bounds β_1, β_2 and regularization parameters μ_1, μ_2 were set to $(\beta_1, \beta_2, \mu_1, \mu_2) = (0.06, 0.002, 4 \times 10^{-4}, 9 \times 10^{-4})$ and the compared simulation results are presented in Fig.5. The L2-error of identified kernels are 0.0046 and 0.0011 using [13] + [9] and our algorithm, respectively, and some other parameters are presented in TABLE III.

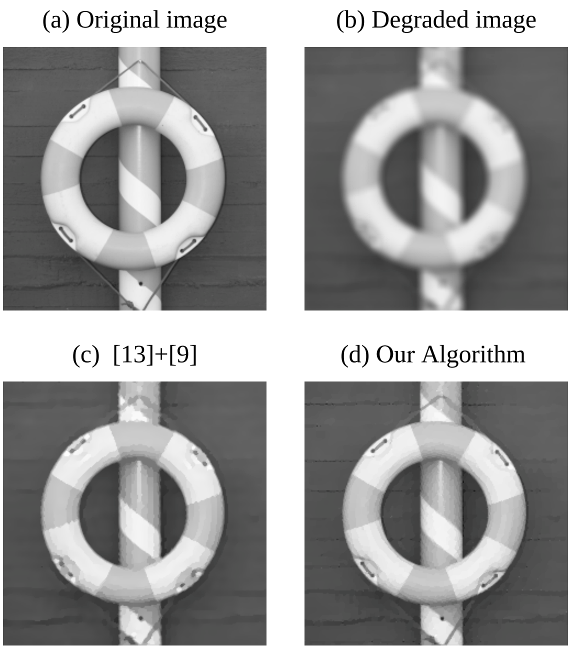

Fig. 5: (a) Original image, (b) blurred image, (c) deblurred by [13] + [9] method ($\lambda_1 = 3 \times 10^{-4}, \lambda_2 = 1 \times 10^{-5}$), (d) output of our algorithm.

C. Blind deblurring of synthetic gray-scale images

Some synthetic images, *Art*, *Zig* and *Sun*, with the same size of 240×240 in Fig.6 were introduced to test the effectiveness for the proposed algorithm. These three images are degraded by 9×9 gaussian, average and disk blurs and additive Gaussian white noise with standard deviation $\sigma = 10^{-4}$, respectively.

The compared restoration results for the above synthetic image *Art* are illustrated in Fig.7 and the simulated results for *Zig* and *Sun* are presented in TABLE IV.

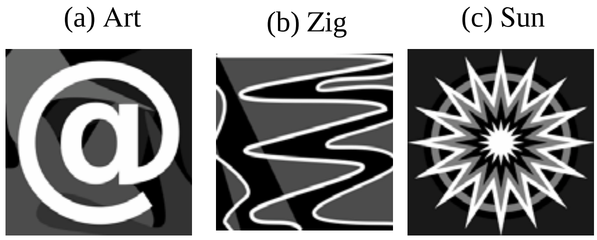

Fig. 6: Different synthetic gray-scale images.

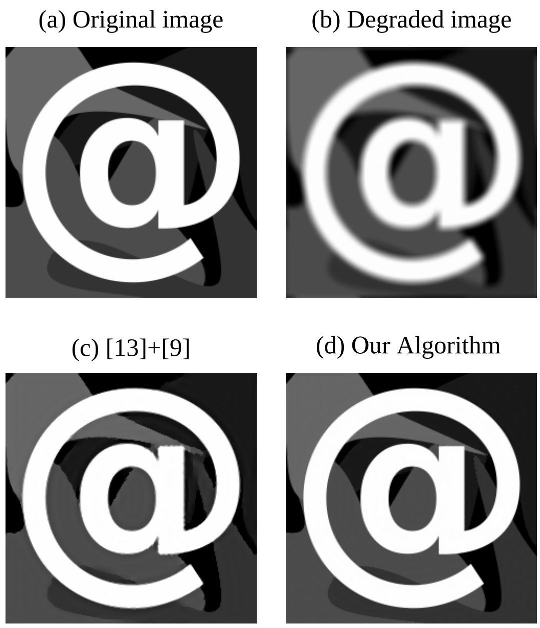

Fig. 7: (a) Original image, (b) degraded image, (c) restored by [13] + [9] method ($\lambda_1 = 3 \times 10^{-4}, \lambda_2 = 2 \times 10^{-5}$), (d) restored by the proposed algorithm.

TABLE IV: Blind deblurring of synthetic gray-scale images: Performance comparison.

PSNR / Blur	PSNR before	PSNR after		SSIM before	SSIM after	
		[13+9]	Proposed		[13+9]	Proposed
G (*Art*)	19.9641	31.7525	**44.1826**	0.7291	0.9713	**0.9979**
A (*Zig*)	14.2461	21.8093	**26.4760**	0.5310	0.7713	**0.8838**
D (*Sun*)	16.9608	19.0682	**30.6138**	0.6834	0.7599	**0.9803**

D. Blind deblurring of color images

Our algorithm was also extended to standard and natural color images. The standard color *Baboon* and *Lena* images (256×256), and *Barbara* (512×512), are imposed by 11×11 Gaussian, Average and Disk blurring kernels and additive Gaussian white noise with $\sigma = 10^{-4}$. The compared results for *Baboon* are shown in Fig.8.

The three different profiles for L2-error of identified kernels are demonstrated in Fig.9 and the three curves present their convergence

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2016.2516949, IEEE Access

TABLE III: Blind deblurring of gray-scale standard images: Performance comparison.

Images (Kernel)	Regulate Parameters of the proposed Algorithm				PSNR/SSIM before	PSNR/SSIM after	
	β_1	β_2	μ_1	μ_2		[13] + [9]	Proposed Algorithm
Lifebuoy (gaussian)	4×10^{-4}	9×10^{-4}	6×10^{-2}	2×10^{-3}	25.9654/0.8272	30.4061/0.8837	**33.7803/0.9173**
Axe (gaussian)	4×10^{-4}	9×10^{-4}	5×10^{-2}	2×10^{-3}	24.9514/0.8510	31.2255/0.9302	**38.6929/0.9783**
Fence (average)	8×10^{-4}	1×10^{-4}	1×10^{-2}	8×10^{-4}	21.1648/0.7423	26.1194/0.8701	**33.6950/0.9613**
Bird (average)	2×10^{-4}	1×10^{-4}	2×10^{-2}	4×10^{-3}	26.9491/0.8839	30.9314/0.9386	**35.4784/0.9753**
Flag (disk)	1×10^{-4}	2×10^{-5}	1×10^{-2}	4×10^{-3}	22.2982/0.7511	23.7515/0.7705	**29.1623/0.8293**
Building (disk)	1×10^{-4}	2×10^{-4}	6×10^{-2}	8×10^{-3}	19.6731/0.4830	20.6847/0.5332	**25.0840/0.7008**

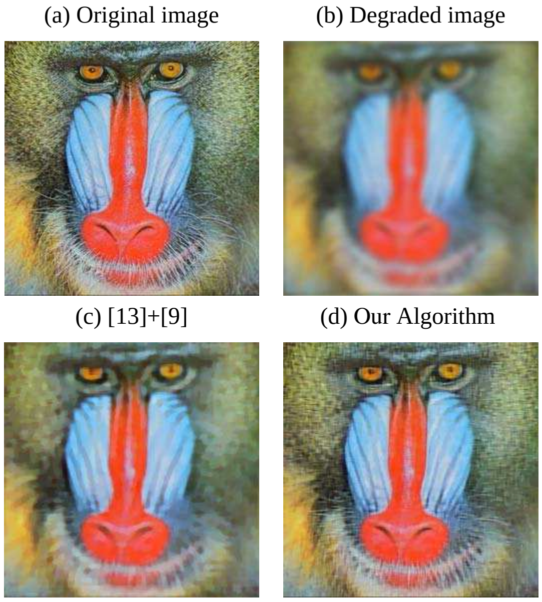

Fig. 8: (a) Original image, (b) degraded image (PSNR=16.3624 dB), (c) restored by [13] + [9] method (PSNR=17.7732 dB, SSIM=0.7149), (d) restored by the proposed algorithm (PSNR=19.6816 dB, SSIM=0.8177).

trend as the iteration proceeds. The simulated results of the below three standard color images about PSNR, SSIM improvements are offered in Table V where the superiority of the proposed algorithm over the [13] + [9] method can readily be observed.

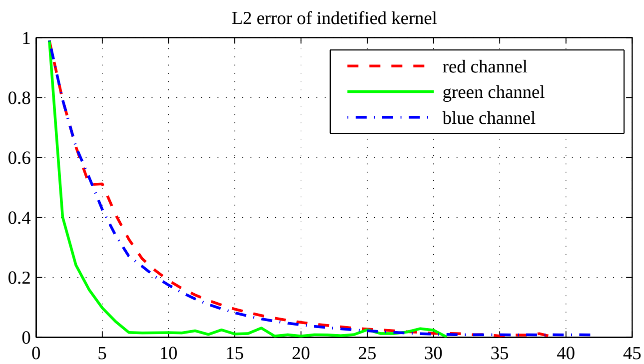

Fig. 9: Different profiles for three channels about L2-Error of identified kernels.

We also report some natural color images restoration results, such as *Flag*, *Fence* and *Lifebuoy*, all these three images with the same size of 240×240. As depicted in Fig.10, each is degraded by a 11×11 blurring kernel and additive Gaussian white noise with $\sigma=10^{-4}$.

TABLE V: Performance comparison of standard color images:

PSNR / Blur	PSNR before	PSNR after		SSIM before	SSIM after	
		[13+9]	Proposed		[13+9]	Proposed
G (*baboon*)	16.3624	17.7732	**19.6816**	0.6386	0.7149	**0.8177**
A (*barbara*)	18.5496	20.3208	**22.0008**	0.7562	0.8068	**0.8966**
D (*lena*)	18.4232	19.2934	**23.4277**	0.9140	0.9196	**0.9652**

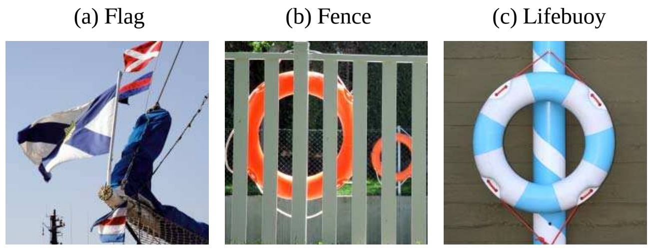

Fig. 10: Different natural color images.

The original, blurred, and deblurred *Fence* images by the two algorithms are shown in Fig.11. The PSNR using our algorithm can improve from 15.5430 dB to **24.7198** dB , [13] + [9] just obtained 18.1674 dB. The other evaluation standard of SSIM can be increased from 0.6266 to **0.9007** offered by the proposed method, but [13] + [9] achieved 0.6983.

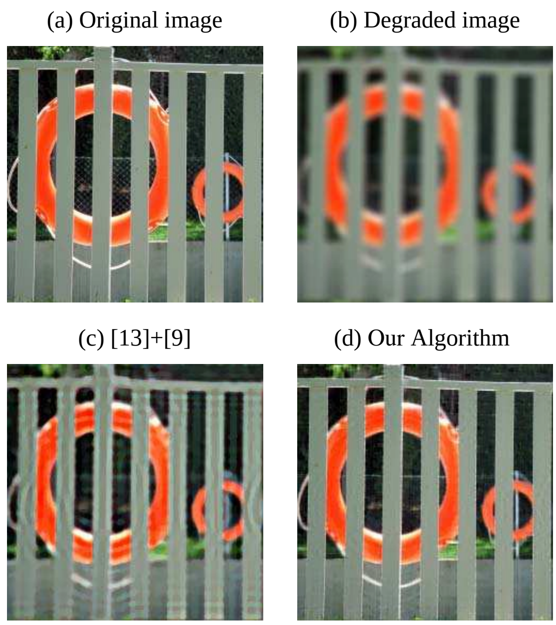

Fig. 11: (a) Original image, (b) degraded image, (c) restored by [13] + [9] method, (d) restored by the proposed algorithm.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2016.2516949, IEEE Access

Comparing the experimental results visually, it is noticed that the proposed algorithm tends to produce images with more detailed information of the textures than [13] + [9]. Except PSNR and SSIM, the detailed information for part of the original and the restored images using two different algorithms are also amplified and demonstrated in Fig.12.

(a) Original (b) [13]+[9] (c) Our Algorithm

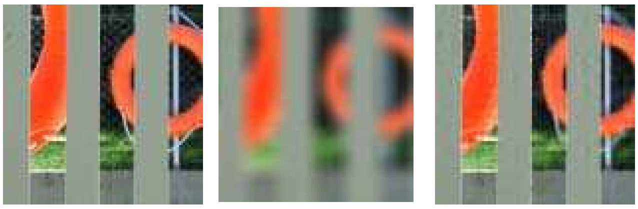

Fig. 12: Close-ups of selected sections for Fig.11 of the original image, restored by [13+9] and our algorithm.

The achievement of the proposed method and [13] + [9] in the aspect of PSNR, SSIM, L2-error of identified kernel and eclipsed time improvements for these three color natural images are given in Table VI. It is obvious that the image information is being remians excellent using the proposed algorithm than [13] + [9] method, especially for the edge information protection from the comparison for Fig. 12 (b) and (c).

TABLE VI: Blind deblurring of natural color *Fence* image: Performance comparison.

Method / *Fence*	[13] + [9]	Proposed Algorithm
SSIM	0.6266→0.6983	**0.6266→0.9007**
PSNR	15.5430→18.1674	**15.5430→24.7198**
L2 Error-Red	0.0034	**0.0021**
L2 Error-Green	0.0034	**0.0021**
L2 Error-Blue	0.0034	**0.0026**
Eclipse Time	56.5200	**33.3839**

The profile of identified kernel's L2-error for natural image *Fence* depicted in Fig. 13 demonstrated the convergence as the iteration proceeded. These three different identified kernel profiles mean that the restored kernel will enormously approach to the original one as the kernel being optimized step by step and finally reach the optimal solutions.

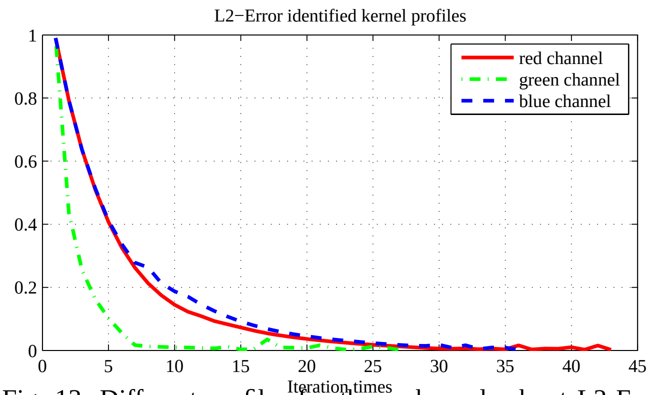

Fig. 13: Different profiles for three channels about L2-Error of identified kernels.

For different natural color images *Fence*, *Flag* and *Lifebuoy*, the improvement of PSNR and SSIM are listed in TABLE VII and from the compared results, we can see that the proposed algorithm demonstrated a better performance than [13] + [9].

TABLE VII: Blind deblurring of natural color images: Performance comparison.

PSNR / Blur	PSNR before	PSNR after		SSIM before	SSIM after	
		[13+9]	Proposed		[13+9]	Proposed
G (*fence*)	15.5430	18.1674	**24.7198**	0.6266	0.6983	**0.9007**
A (*flag*)	15.7676	18.9462	**26.4884**	0.7720	0.8431	**0.9653**
D (*lifebuoy*)	20.0350	21.6604	**26.5644**	0.8907	0.9095	**0.9521**

V. Conclusion

A fast weighted algorithm for blind deconvolution removing unknown blurs as well as random noise is proposed. This methodology can synchronously restore image and identify kernel as can not be achieved by other recent algorithms. The performance of the improved algorithm relative to well-established deblurring methods in the literature has been demonstrated by applying it to a variety of gray-scale standard, natural, synthetic images and color images. In the evaluation standard about PSNR, SSIM and convergence time improvement, the proposed algorithm achieved better results than other algorithms.

References

[1] L. I. Rudin, S. Osher, and E. Fatemi. Nonlinear total variation based noise removal algorithms. Physica D: Nonlinear Phenomena, 60(1-4):259–268, 1992.

[2] L. I. Rudin and S. Osher. Total variation based image restoration with free local constraints. In Image Processing, 1994. Proceedings. ICIP-94., IEEE International Conference, volume 1, pages 31–35. IEEE, 1994.

[3] C. R. Vogel and M. E. Oman. Fast, robust total variation-based reconstruction of noisy, blurred images. Image Processing, IEEE Transactions on, 7(6):813–824, 1998.

[4] M. K. P. Ng, R. H. Chan, and W. C. Tang. A fast algorithm for deblurring models with neumann boundary conditions. SIAM Journal on Scientific Computing, 21(3), 1999.

[5] A. Beck and M. Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. SIAM Journal on Imaging Sciences, 2(1):183–202, 2009.

[6] H.-Y. Liu,, W.-S. Lu, and M. Q. H. Meng. De-blurring wireless capsule endoscopy images by total variation minimization. In Communications, Computers and Signal Processing (PacRim), 2011 IEEE Pacific Rim Conference on, pages 102–106. IEEE, 2011.

[7] H.-Y. Liu, W.-S. Lu, C.-J. Zhang, and Q.-H. Meng. Research and application for fast restoration of color images algorithm based on total variation framework. In Journal Beijing University of Post and Telecommunications, 35(6), pages 1–5. 2012.

[8] Zh. Bai, C. Daniele, D. Marco and S. Stefano. A fast alternating minimization algorithm for total variation deblurring without boundary artifacts. In Journal of Mathematical Analysis and Applications, 415(1), pages 373–393. IEEE, 2014.

[9] A. Beck and M. Teboulle. Fast gradient-based algorithms for constrained total variation image denoising and deblurring problems. IEEE Transactions on Image Processing, 18(11):2419–2434, 2009.

[10] Y. L. You and M. Kaveh. A regularization approach to joint blur identification and image restoration. IEEE Transactions on Image Processing, 5(3):416–428, 1996.

[11] Y. L. You and M. Kaveh. Blind image restoration by anisotropic regularization. IEEE Transactions on Image Processing, 8(3):396–407, 1999.

[12] L. He, A. Marquina, and S. J. Osher. Blind deconvolution using tv regularization and bregman iteration. International Journal of Imaging Systems and Technology, 15(1):74–83, 2005.

[13] T. F. Chan and C. K. Wong. Total variation blind deconvolution. IEEE Transactions on Image Processing, 7(3):370–375, 1998.

[14] M. S. C. Almeida and L. B. Almeida. Blind and semi-blind deblurring of natural images. IEEE Transactions on Image Processing, 19(31):36–52, 2010.

[15] T. F. Chan and C. K. Wong. Convergence of the alternating minimization algorithm for blind deconvolution. Linear Algebra and its Applications, 316(1-3):259–285, 2000.

[16] P. Getreuer. Total variation deconvolution using split Bregman. Image Processing OnLine, 2(1):158–174, 2012.

[17] K. Ohkoshi, T. Goto, S. Hirano and M. Sakurai. Blind image restoration based on total variation regularization of blurred images. 2012 IEEE 1st Global Conference on Consumer Electronics, 621–622, 2012.

[18] C. He , C. Hu, W. Zhang, B. Shi and X. Hu. Fast total-variation image deconvolution with adaptive parameter estimation via split Bregman method. Mathematical Problems in Engineering, 2014.

[19] E. J. Candès, J. Romberg and T. Tao. Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information. IEEE Transactions on Information Theory, 52(2):489–509, 2006.

[20] L. Fan, C. J. Louis, J. Wu and H. Shu. A New Fast Algorithm for Constrained Four-Directional Total Variation Image Denoising Problem. Mathematical Problems in Engineering, 501:1–11, 2014.

[21] M. F. Fahmy, G. M. Abdel Raheem, U. S. Mohammed and O. F. Fahmy. A new total variation based image denoising and deblurring technique. Europe Conference, 1669–1675, 2013.

[22] Z. Zuo , T. Zhang, X. Lan and L. Yan. An adaptive non-local total variation blind deconvolution employing split Bregman iteration. Circuits, Systems, and Signal Processing, 32(5):2407–2421, 2013.

[23] A. Ahmed, B. Recht, J. Romberg. Blind deconvolution using convex programming. IEEE Transactions on Information Theory, 60(3):1711–1732, 2014.

[24] K. Shao, Y. Zou, Y. Liu, C. Li and B. Fu. Based on Total Variation Regularization Iterative Blind Image Restoration Algorithm. Sensors and Transducers, 167(3):36–42, 2014.

[25] D. Krishnan, J. Bruna and R. Fergus. Blind deconvolution with reweighted sparsity promotion. CoRR abs, 60(3):1–16, 2013.

[26] L. Yan, H. Fang and S. Zhong. Blind image deconvolution with spatially adaptive total variation regularization. Optics letters, 37(14):2778–2780, 2012.

[27] F. Sroubek and P. Milanfar. Robust multichannel blind deconvolution via fast alternating minimization. IEEE Transactions on Image Processing, 21(4):1687–1700, 2012.

[28] G. Gong, H. Zhang and M. Yao. Construction model for total variation regularization parameter. Optics express, 22(9):10500–10508, 2014.

[29] S. Tang, W. Gong, W. Li and W. Wang. Non-blind image deblurring method by local and nonlocal total variation models. Signal Processing, 94:339–349, 2014.

[30] F. Dong, H. Zhang and D. Kong. Nonlocal total variation models for multiplicative noise removal using split Bregman iteration. Mathematical and Computer Modelling, 55(3):939–954, 2012.

[31] P. Rodríguez. Total variation regularization algorithms for images corrupted with different noise models: a review. Journal of Electrical and Computer Engineering, 2013:1–18, 2013.

[32] D. Perrone and P. Favaro. Total variation blind deconvolution: The devil is in the details. IEEE Conference on Computer Vision and Pattern Recognition, 2909–2916, 2014.

[33] M. Renu, S. Chaudhuri and R. Velmurugan. Convergence analysis of a quadratic upper bounded TV regularizer based blind deconvolution. Signal Processing, 106, 174–183, 2015.

[34] A. Kheradmand, and P. Milanfar. A general framework for regularized, similarity-based image restoration. IEEE Transactions on Image Processing, 23(12):5136–5151, 2014.

[35] G. Peyre, S. Bougleux, and L. D. Cohen. Non-local regularization of inverse problems. Inverse Problems and Imaging, 5(2):511–530, 2011.

[36] A. Danielyan, V. Katkovnik, and K. Egiazarian. BM3D frames and variational image deblurring. IEEE Transactions on Image Process, 21(4):1715–1728, 2012.

Haiying Liu Dr. Haiying Liu is currently conducting research as a postdoctoral fellow in the Department of Electrical and Computer Engineering, Dalhousie University, Canada. She received her M. Sc. and Ph.D. degrees from Shandong University, School of Control Science and Engineering. She was a visiting student from August 2009 to August 2011 in the Department of Electrical and Computer Engineering, University of Victoria, Canada. Her research interests are mainly in the field of Pattern Recognition, Digital Image Processing and Computer Vision, adaptive fuzzy controller design and stability analysis with application in mobile robots.

Jason Gu Dr. Jason Gu received his bachelor's degree from Electrical Engineering and Information Science at the University of Science and Technology of China in 1992, his Master's degree from Biomedical Engineering at Shanghai Jiaotong University in 1995, and PhD degree from University of Alberta in Canada in 2001. He is currently a full professor in Electrical and Computer Engineering at Dalhousie University in Canada. He is also a cross-appointed professor in School of Biomedical Engineering for his multidisciplinary research work. Dr. Gu's research areas include robotics, biomedical engineering, rehabilitation engineering, neural networks, and control. He has over nineteen years research and teaching experience and has published over 240 conference papers and articles. He has been the associate editor for the following journals: Journal of Control and Intelligent Systems, Transactions on CSME, Canada IEEE Transaction on Mechatronics, International Journal of Robotics and Automation, Unmanned Systems, Journal of Engineering and Emerging Technologies and IEEE Access. Dr. Gu is a Fellow of the Engineering Institute of Canada.

Max Q.-H. Meng Professor Meng received his Ph.D. degree in Electrical and Computer Engineering at the University of Victoria in Canada in 1992, following his M.Sc. degree in Automatic Control at the Beijing Institute of Technology in Beijing, China in 1988. Max was a Professor in the Department of Electrical and Computer Engineering at the University of Alberta in Canada from April 1994 to August 2004. Currently Max is a Professor in the Department of Electronic Engineering at the Chinese University of Hong Kong. Professor Meng is a Fellow of IEEE.

Wu-Sheng Lu Dr. Wu-Sheng Lu received his M.S. degree in electrical engineering, and Ph.D. degree in control science from the University of Minnesota, Minneapolis, USA, in 1983 and 1984, respectively. He was a post-doctoral fellow at the University of Victoria, Victoria, B.C., Canada, in 1985, and a visiting Assistant Professor at the University of Minnesota from January 1986 to April 1987. He joined the Electrical and Computer Engineering Department, University of Victoria, in May 1987 as an Associate Professor, and has been a Professor there since 1991. His current research interests include analysis and design of digital filters, digital signal and image processing with a focus on sparse signal processing, and methods and applications of convex optimization. He is the co-author with A. Antoniou of Two-Dimensional Digital Filters and Practical Optimization: Algorithms and Engineering Applications. He is a Fellow of the Engineering Institute of Canada and IEEE. He became a Life Fellow of the IEEE in 2012.