JOINT DEBLURRING AND DEMOSAICING OF POISSONIAN BAYER-DATA BASED ON LOCAL ADAPTIVITY

Dmytro Paliy[a⋆], Alessandro Foi[a], Radu Bilcu[b], Vladimir Katkovnik[a], and Karen Egiazarian[a]

[a]Department of Signal Processing, Tampere University of Technology, P.O. Box 553, FIN-33101, Tampere, Finland.
web: www.cs.tut.fi/∼lasip e-mail: firstname.lastname@tut.fi
[b]Nokia Research Center, Tampere, Finland. e-mail: firstname.lastname@nokia.com

ABSTRACT

We present a novel technique for joint deblurring and demosaicing of noisy Poissonian Bayer data (e.g., data acquired by a digital CMOS or CCD imaging sensor). The technique incorporates the regularized inverse and the Wiener inverse with adaptive filtering based on the concept of cross-color local polynomial approximation (LPA) and intersection of confidence intervals (ICI). The directional filters designed by LPA utilize simultaneously the green, red, and blue color components. This is achieved by a linear combination of complementary-supported smoothing and derivative kernels designed for the Bayer data grid. The ICI rule is used for data-adaptive selection of the length of the designed cross-color directional filter. Simulation experiments demonstrate the efficiency of the proposed technique with respect to the conventional approach where deconvolution and demosaicing are computed independently.

1. INTRODUCTION

In single-chip digital imaging systems the light passes through the optical system of the camera and is focused at the digital sensor, where it is acquired through a color filter array (CFA). However, due to various optical distortions (e.g., out-of-focus blur, motion, etc.), the image is blurred before the CFA sampling. Since all significant sources of noise are within the sensor (each site works as a photon-counter, thus the noise is predominantly signal-dependant and Poissonian), it results in the acquisition of noisy downsampled blurred data, with these three degradations (blur, noise, and sampling) realized in this given order. Because the blurring comes before the downsampling, the deblurring of this data is particularly ill-posed. Thus, the problem of restoring a full-color full-size image from the acquired data becomes much more difficult than the CFA interpolation where there is only noise and no blur.

The research around this more general and complex problem is rather recent, and thus there exist only a very limited number of works dedicated to its solution. The mathematical modelling and solution based on a quadratic criterion is discussed in [1]. Essential progress in this approach has been demonstrated in [2] and [3], where the problem is considered within the super-resolution imaging framework (thus, allowing multiple images to be combined in the interpolation).

In these works, the criterion functionals include the fidelity terms and penalties to enable both the smoothness of the color solutions as well as some color correlation. In particular, the prior used in [3] forces the red, green, and blue high-frequency components to be mutually similar. The relative weights of the similarities between different color components are modulated by weights and a recursive procedure is developed for fitting these weights to the observations.

Overall, the approach proposed in the recent papers [2] and [3] gives universal formulation of the problem and recipes for the algorithm development. However, it suffers from limitations typical for general formulation of imaging as a global optimization problem, which needs to be solved by recursive algorithms.

Our contribution has a pragmatic goal and aims at avoiding these limitations following the fact that for the separate deblurring and CFA interpolation there exist already very efficient and powerful nonrecursive algorithms. Exploiting these algorithms, we design a novel technique for *joint* deblurring and demosaicing. In particular, the proposed technique is based on cross-color filtering [4] and on a two-stage regularized deconvolution [5], specifically designed for Poissonian data. This approach is of practical interest for implementation in devices with limited power and computational resources and shows results competitive with other more computationally expensive deblurring/interpolation methods.

The rest of the paper is organized as follows. In Section 2 we consider a mathematical model used for the image acquisition process and two general approaches to solve the related inverse problem. In Section 3 we derive an approximate point-spread function (PSF) for deconvolution of subsampled color components and introduce relevant upsampling and downsampling operators. The proposed joint deblurring and demosaicing algorithm is described in Section 4. The simulation results and analysis is given in Section 5.

2. OBSERVATION MODEL FOR BLURRED NOISY SAMPLED DATA

Let $y_{RGB}(x) = (y_R(x), y_G(x), y_B(x))$ be the true color image and $\tilde{y}_{RGB}(x) = (\tilde{y}_R(x), \tilde{y}_G(x), \tilde{y}_B(x))$ be the blurred color image with the same PSF υ for all color components[1] where

$$\tilde{y}_c(x) = (\upsilon \circledast y_c)(x), \quad x \in X, \quad c = R, G, B.$$

Here, the vector maps of blurred ($\tilde{y}_{RGB}$) and true (y_{RGB}) color images are defined on the full-size $N \times M$ grid X (N and M are even).

The Bayer mask observations assume the sampling of the blurred image $\tilde{y}_{RGB}(x)$ according to the structure of the Bayer color sensors [6]. These sampled signals can be given in the form

$$z_{\text{bayer}}(x) = \mathcal{B}\{\tilde{y}_{RGB}\}(x) + \sigma(x)n(x), \tag{1}$$

where n is a white random zero-mean noise with variance equal to one and $\mathcal{B}$ is the Bayer sampling operator. This operator is conventionally defined as follows:

$$\mathcal{B}\{\tilde{y}_{RGB}\}(x) = \begin{cases} \tilde{y}_R(x), & x \in X_R, \\ \tilde{y}_G(x), & x \in X_{G_1} \cup X_{G_2}, \\ \tilde{y}_B(x), & x \in X_B, \end{cases} \tag{2}$$

where X_{G_1}, X_{G_2}, X_R, X_B are the grids of two greens, red, and blue color sensor arrays,

$$\begin{aligned} X_{G_1} &= \{(x_1,x_2) : x_1 = 1,3,\ldots,N-1,\ x_2 = 1,3,\ldots,M-1\}, \\ X_{G_2} &= \{(x_1,x_2) : x_1 = 2,4,\ldots,N,\ x_2 = 2,4,\ldots,M\}, \\ X_R &= \{(x_1,x_2) : x_1 = 1,3,\ldots,N-1,\ x_2 = 2,4,\ldots,M\}, \\ X_B &= \{(x_1,x_2) : x_1 = 2,4,\ldots,N,\ x_2 = 1,3,\ldots,M-1\}. \end{aligned}$$

This work was supported in part by the Finnish Funding Agency for Technology and Innovation (Tekes - AVIPA2 project) and by the Academy of Finland (application no. 213462, Finnish Programme for Centres of Excellence in Research 2006-2011, and application no. 118312, Finland Distinguished Professor Programme 2007-2010).

⋆ Contact author. e-mail: dmitriy.paliy@tut.fi

[1] This restriction is not essential: the proposed algorithm can be used also with different PSF for different color components.

The dependence of the noise on the signal is given by $\sigma(x)$, which is a deterministic function of the image intensity. In particular, we consider Poissonian distributed noise with

$$\chi z_{\text{bayer}}(x) \sim \mathcal{P}(\chi \mathcal{B}\{\tilde{y}_{RGB}\}(x)),\ x \in X, \tag{3}$$

where $\chi > 0$ is a noise scaling parameter and $\mathcal{P}$ denotes the Poissonian distribution. This noise can be written explicitly in the additive form (1) where the variance depends on the image intensity as $\sigma^2(x) = \text{var}\{z_{\text{bayer}}(x)\} = \mathcal{B}\{\tilde{y}_{RGB}\}(x)/\chi$. Our goal is to reconstruct the true image y_{RGB} from z_{bayer}.

In order to discuss the idea of our approach and the structure of the developed algorithm, we introduce generic deblurring and interpolation operators $\mathcal{D}$ and $\mathcal{I}$. We assume that the output of the deblurring operator has the same domain as the input. Thus, we can write both $\mathcal{D}(\tilde{y}_R) = \hat{y}_R$ and $\mathcal{D}(\tilde{y}_{R,\text{sub}}) = \hat{y}_{R,\text{sub}}$, where the subscript "sub" denotes the subsampling of a full-size color plane on the respective small-size subgrid X_{G_1}, X_{G_2}, X_R, or X_B, i.e. $\tilde{y}_{R,\text{sub}} : X_R \to \mathbb{R}$, $\tilde{y}_{R,\text{sub}}(x) = \tilde{y}_R(x)$, $x \in X_R$. Analogously, the operator $\mathcal{I}$ interpolates the input data on its respective small-size subgrid; for instance, the reconstruction of the full-size red signal y_R from the subsampled $y_{R,\text{sub}}$ can be written as $\mathcal{I}(y_{R,\text{sub}}) = \hat{y}_R$.

Formally, the two operators can be combined either as $\mathcal{I}(\mathcal{D}(\tilde{y}_{R,\text{sub}})) = \hat{y}_R$ or as $\mathcal{D}(\mathcal{I}(\tilde{y}_{R,\text{sub}})) = \hat{y}_R$. However, it is well known that there is no chance to get a good quality for color reconstruction using subsampled data of a single color only; the color correlation is the key ingredient of all the successful demosaicing algorithms. Furthermore, because of aliasing, it is not possible to model the blur in $\tilde{y}_{R,\text{sub}}$ as a convolution operation of $y_{R,\text{sub}}$ against a given PSF. Thus, the deblurring on a single subgrid is severely ill-posed. To exploit the color correlation, a CFA interpolation operator $\mathcal{I}_{\text{CFAI}}$ is introduced, such that $\mathcal{B}\{\mathcal{I}_{\text{CFAI}}(\mathcal{B}\{\tilde{y}_{RGB}\})\} = \mathcal{B}\{\tilde{y}_{RGB}\}$ and $\mathcal{B}\{\mathcal{I}_{\text{CFAI}}(\mathcal{B}\{y_{RGB}\})\} = \mathcal{B}\{y_{RGB}\}$.

Using the introduced operators, the structure of the reconstruction algorithms can be explained in a simple and clear form:

- *Deblurring-Interpolation algorithms*

$$\mathcal{I}_{\text{CFAI}}\big(\mathcal{D}(\tilde{y}_{R,\text{sub}}) + \mathcal{D}(\tilde{y}_{G,\text{sub}}) + \mathcal{D}(\tilde{y}_{B,\text{sub}})\big) = (\hat{y}_R, \hat{y}_G, \hat{y}_B), \tag{4}$$

- *Interpolation-Deblurring algorithms*

$$\mathcal{I}_{\text{CFAI}}(\tilde{y}_{R,\text{sub}} + \tilde{y}_{G,\text{sub}} + \tilde{y}_{B,\text{sub}}) = (\tilde{y}_R, \tilde{y}_G, \tilde{y}_B), \tag{5}$$
$$\mathcal{D}(\tilde{y}_R) = \hat{y}_R,\ \mathcal{D}(\tilde{y}_G) = \hat{y}_G,\ \mathcal{D}(\tilde{y}_B) = \hat{y}_B.$$

In the summations (4)-(5) we tacitly assumed that the subsampled data are zero-padded outside their respective subgrids of definition. In this paper we follow (4) because it is more computationally efficient and (5) is a subject for a future work. The naive straightforward approach would be to apply first a deconvolution $\mathcal{D}(\cdot)$ designed for Poissonian data (e.g., [7],[8]) and then use a CFA interpolation $\mathcal{I}_{\text{CFAI}}(\cdot)$ assuming that the input is deblurred and noise free. Instead, we combine the deconvolution and demosaicing into a single procedure.

3. PRELIMINARIES

3.1 Downsampling and upsampling operators for the Bayer grid and for convolution kernels

Before we proceed, we define some downsampling and upsampling operators. Let P and Q be two generic data arrays:

$$P = \begin{bmatrix} a & b & c & d & e & \cdot \\ f & g & h & i & j & \\ k & l & m & n & o & \\ p & q & r & s & t & \\ \cdot & & & & & \cdot \end{bmatrix}, \quad Q = \begin{bmatrix} a & b & c & \cdot \\ d & e & f & \\ \cdot & & & \cdot \end{bmatrix}.$$

The *Bayer downsampling* operators are naturally defined as

$$\downarrow_{G_1}\{P\} = \begin{bmatrix} a & c & e & \cdot \\ k & m & o & \\ \cdot & & & \cdot \end{bmatrix}, \quad \downarrow_R\{P\} = \begin{bmatrix} b & d & \cdot \\ l & n & \\ \cdot & & \cdot \end{bmatrix},$$

$$\downarrow_B\{P\} = \begin{bmatrix} f & h & j & \cdot \\ p & r & t & \\ \cdot & & & \cdot \end{bmatrix}, \quad \downarrow_{G_2}\{P\} = \begin{bmatrix} g & i & \cdot \\ q & s & \\ \cdot & & \cdot \end{bmatrix}.$$

Analogously, the *Bayer upsampling* operators are defined as

$$\uparrow_{G_1}\{Q\} = \begin{bmatrix} a & 0 & b & 0 & \cdot \\ 0 & 0 & 0 & 0 & \\ c & 0 & d & 0 & \\ 0 & 0 & 0 & 0 & \\ \cdot & & & & \cdot \end{bmatrix}, \quad \uparrow_R\{Q\} = \begin{bmatrix} 0 & a & 0 & b & \cdot \\ 0 & 0 & 0 & 0 & \\ 0 & c & 0 & d & \\ 0 & 0 & 0 & 0 & \\ \cdot & & & & \cdot \end{bmatrix},$$

$$\uparrow_B\{Q\} = \begin{bmatrix} 0 & 0 & 0 & 0 & \cdot \\ a & 0 & b & 0 & \\ 0 & 0 & 0 & 0 & \\ c & 0 & d & 0 & \\ \cdot & & & & \cdot \end{bmatrix}, \quad \uparrow_{G_2}\{Q\} = \begin{bmatrix} 0 & 0 & 0 & 0 & \cdot \\ 0 & a & 0 & b & \\ 0 & 0 & 0 & 0 & \\ 0 & c & 0 & d & \\ \cdot & & & & \cdot \end{bmatrix}.$$

Given two generic kernels

$$g_{\text{big}} = \begin{bmatrix} & & \cdot & & \\ & a & b & c & d & e \\ & f & g & h & i & j \\ \cdot & k & l & [m] & n & o & \cdot \\ & p & q & r & s & t \\ & u & v & w & x & y \\ & & & \cdot & & \end{bmatrix}, \quad g_{\text{small}} = \begin{bmatrix} & & \cdot & & \\ & a & b & c & \\ \cdot & d & [e] & f & \cdot \\ & g & h & i & \\ & & \cdot & & \end{bmatrix},$$

where the brackets [ ] denote the origin of the kernel, we define four *kernel downsampling operators*

$$\downarrow_{K_0}\{g_{\text{big}}\} = \begin{bmatrix} & & \cdot & & \\ & a & c & e & \\ \cdot & k & [m] & o & \cdot \\ & u & w & y & \\ & & \cdot & & \end{bmatrix}, \quad \downarrow_{K_1}\{g_{\text{big}}\} = \begin{bmatrix} & & \cdot & & \\ & b & d & & \\ \cdot & [l] & n & \cdot & \\ & v & x & & \\ & & \cdot & & \end{bmatrix},$$

$$\downarrow_{K_2}\{g_{\text{big}}\} = \begin{bmatrix} & & \cdot & & \\ & f & [h] & j & \\ \cdot & p & r & t & \cdot \\ & & \cdot & & \end{bmatrix}, \quad \downarrow_{K_3}\{g_{\text{big}}\} = \begin{bmatrix} & & \cdot & & \\ & [g] & i & & \\ \cdot & q & s & \cdot & \\ & & \cdot & & \end{bmatrix}.$$

and four *kernel upsampling operators* $\uparrow_{K_i}$, $i = 0, 1, 2, 3$,

$$\uparrow_{K_i}\{g_{\text{small}}\} = \begin{bmatrix} & & & \cdot & & & \\ & a & 0 & b & 0 & c & \\ & 0 & 0 & 0 & 0 & 0 & \\ \cdot & d & 0 & [e] & \{0\} & f & \cdot \\ & 0 & 0 & (0) & \langle 0 \rangle & 0 & \\ & g & 0 & h & 0 & i & \\ & & & \cdot & & & \end{bmatrix},$$

where the brackets [], {}, (), and $\langle\rangle$ indicate the origin of the upsampled kernel $\uparrow_{K_0}\{g_{\text{small}}\}$, $\uparrow_{K_1}\{g_{\text{small}}\}$, $\uparrow_{K_2}\{g_{\text{small}}\}$, and $\uparrow_{K_3}\{g_{\text{small}}\}$, respectively. Note that the above four upsampling operators are different only for what concerns the location of the origin in the output kernel. Also observe that the origin seems to be shifted to the direction opposite to the expected one. This is due to the change of sign of the argument within the convolution operators.

3.2 Downscaled PSF for Bayer subcomponents

The Deblurring-Interpolation algorithms (4) require that a "downscaled version" v_{sub} of the PSF v is used to deconvolve the downsampled color components. In general, because of aliasing, it is not possible to model the blurring in the Bayer subcomponents as a standard convolutional blur. Nevertheless, it is very practical and desirable to use a convolutional approximation for the blur operator, since it gives the opportunity to use very efficient deblurring procedures. The accuracy of the convolutional approximation depends on the downscaled PSF v_{sub} used for in the convolution model. An "optimal" PSF v_{sub} is essentially image-dependant because of aliasing. However, in practice, the downscaled v_{sub} should be determined from v only. This becomes possible provided that some natural restrictions are imposed on y.

Let y be a band-limited image of size $N \times M$ with only the $\frac{N}{2} \times \frac{M}{2}$ (in practice $\frac{N}{2} - 1 \times \frac{M}{2} - 1$, to keep the data real) non-zero low-frequency harmonics (around the central DC term). In what follows we will refer to coefficients that fall inside or outside this band as low-frequency or high-frequency terms, respectively. We denote by $\mathcal{F}$ and $\mathcal{F}_{\text{sub}}$, respectively, the Fourier transforms for a domain of size $N \times M$ and of size $\frac{N}{2} \times \frac{M}{2}$.

Ignoring the presence of noise and omitting the normalizations of the Fourier transform, in frequency domain we have that $Z = YV$, where Z, Y, and V are the $\mathcal{F}$ transforms of z, y, and v, respectively. However, since Y is low-pass, the above equation depends only on the low-frequency portion of the spectrum of V, i.e. $Z = YV = YV_{\text{lp}}$, where V_{lp} is a low-pass version of V that has all high-frequency terms equal to zero.

Let now $z = \mathcal{F}^{-1}(YV)$ and consider a generic Bayer downsampled version of z, denoted as z_{sub}. Similarly, we define y_{sub} as the downsampled y. Both y and z are low-pass, therefore the downsampling can be reversed without loss of information as there is no aliasing. Moreover, it is possible to formally deconvolve z_{sub} and obtain both y_{sub} and y. More precisely we have that,

$$y_{\text{sub}} = \mathcal{F}_{\text{sub}}^{-1}\left(\frac{\mathcal{F}_{\text{sub}}(z_{\text{sub}})}{V_{\text{sub}}}\right), \tag{6}$$

where V_{sub} is the $\frac{N}{2} \times \frac{M}{2}$ central portion the spectrum of V_{lp}.

The downscaled PSF is defined as $\mathcal{F}_{\text{sub}}^{-1}(V_{\text{sub}})$. In case V_{sub} has zeros, a pseudo- or regularized inverse can be used in place of the naive inversion in (6).

Analogously, by zero-padding the high-frequencies in Fourier domain (denoted by the operator $\mathbf{0}_{\text{pad}}$), we obtain

$$y = \mathcal{F}^{-1}\left(\mathbf{0}_{\text{pad}}\left(\frac{\mathcal{F}_{\text{sub}}(z_{\text{sub}})}{V_{\text{sub}}}\right)\right).$$

The above equality assumes that either a modulation on the spectrum or (equivalently) a circular shift after inverse-transformation is performed whenever the downsampling starts from an even sample (e.g., $[2:2:N] \times [2:2:N]$).

We remark that this PSF modeling (i.e., assuming that y is band-limited) is approximate, since the image y has usually a full spectrum and thus aliasing would play a role. However, the proposed simplified modeling allows to establish a formally correct setting for the problem, where a direct relation between the full-size PSF and the downscaled PSF exists.

Because of aliasing, artifacts in general appear, at least to some extent, when the color subcomponents z_{sub} are deconvolved as in (6). Nevertheless, we wish to emphasize that since the observations are blurred, the aliasing is not as strong as in the case of CFA interpolation of blur-free data. Moreover, the filtering which is used in our deblurring approach is spatially adaptive and can compensate for potential undesirable artifacts, as shown in [4].

4. ALGORITHM

In this section we describe in detail the proposed algorithm. Its block diagram is given in Figure 1.

4.1 Regularized inverse

The regularized inverse (RI) linear filter is applied separately to each observed color subcomponent R, G_1, G_2, and B. It is expressed in the frequency domain as

$$\hat{Y}_c^{\text{RI}} = \Phi_{\text{sub}}^{\text{RI}} Z_c, \quad c = R, G_1, G_2, B, \qquad \Phi_{\text{sub}}^{\text{RI}} = \frac{V_{\text{sub}}^*}{|V_{\text{sub}}|^2 + \varepsilon_{\text{RI}}^2}, \tag{7}$$

where V_{sub} is a downscaled PSF obtained from v as described in Section 3.2, $\varepsilon_{\text{RI}}^2$ is a regularization parameter, and Z_c corresponds to Fourier transform $\mathcal{F}_{\text{sub}}$ of the sampled observations z_c, i.e.

$$Z_c = \mathcal{F}_{\text{sub}}\{z_c\}, \quad z_c = \downarrow_c \{z_{\text{bayer}}\}, \quad c = R, G_1, G_2, B.$$

We denote the RI operator in the spatial domain as $\varphi_{\text{sub}}^{\text{RI}} = \mathcal{F}_{\text{sub}}^{-1}\{\Phi_{\text{sub}}^{\text{RI}}\}$. We emphasize that $\varphi_{\text{sub}}^{\text{RI}}$ has the size of a subsampled signal, since it is defined using the $\mathcal{F}_{\text{sub}}^{-1}$ inverse transform. Four deconvolved estimates, one for each color subcomponent of the Bayer pattern, are obtained by the inverse Fourier transform of (7):

$$\hat{y}_c^{\text{RI}} = \mathcal{F}_{\text{sub}}^{-1}\left(\hat{Y}_c^{\text{RI}}\right) = \mathcal{F}_{\text{sub}}^{-1}(\Phi_{\text{sub}}^{\text{RI}} Z_c) = \varphi_{\text{sub}}^{\text{RI}} \circledast z_c, \quad c = R, G_1, G_2, B.$$

While in these estimates the blur is reduced, they are still rather noisy because only a mild regularization is used in (7). In this sense, we may consider these estimates as subsampled components of a noisy blur-free image. These components are combined together in a full-size image as

$$\hat{y}_{\text{bayer}}^{\text{RI}} = \uparrow_R \{\hat{y}_R^{\text{RI}}\} + \uparrow_{G_1} \left\{\hat{y}_{G_1}^{\text{RI}}\right\} + \uparrow_{G_2} \left\{\hat{y}_{G_2}^{\text{RI}}\right\} + \uparrow_B \{\hat{y}_B^{\text{RI}}\}.$$

Thus, an estimate of the blur-free and noise-free image y can be obtained from $\hat{y}_{\text{bayer}}^{\text{RI}}$ exploiting techniques for interpolation of *noisy* Bayer-data. However, because of the RI filter, the noise in each of the components $\hat{y}_c^{\text{RI}}$ is *colored*. Therefore, in order to remove this noise accurately, one needs to take into account the impulse response of the deblurring filter. In this sense, computing the deblurring and the interpolation as two independent steps is unnatural. Instead, these two operations need to be combined as a joint filtering procedure. Moreover, because of the possible aliasing in $\hat{y}_c^{\text{RI}}$, the interpolation procedure must be especially accurate in removing artifacts.

For these purposes, we rely on a modified version of the LPA-ICI algorithm for interpolation of noisy Poissonian Bayer data [4]. It is applied to the Bayer pattern. The developed modification allows the algorithm to exploit the specific statistics of the deblurred color subcomponents and, in particular, the nonuniformity of the variance across the Fourier spectrum. Overall, the proposed procedure can be interpreted as an interpolation algorithm based on directional adaptive-scale cross-color kernels that embeds a deblurring RI filter.

The algorithm processes the Bayer data $\hat{y}_{\text{bayer}}^{\text{RI}}$ by convolutions against varying-scale directional cross-color LPA *kernels* $g_{h,\theta}$ [4], with the corresponding *estimates* $\hat{y}_{h,\theta}^{\text{RI}}$ computed as

$$\hat{y}_{h,\theta}^{\text{RI}} = \hat{y}_{\text{bayer}}^{\text{RI}} \circledast g_{h,\theta}, \tag{8}$$

with the parameters θ and h indicating the direction and length of the kernel $g_{h,\theta}$. Denoised and interpolated estimates for the different colors are found at different spatial positions within the cross-color estimate $\hat{y}_{h,\theta}^{\text{RI}}$.

In general and without loss of generality, the kernel $g_{h,\theta}$ can be decomposed into four kernels $g_{h,\theta}^{K_0}$, $g_{h,\theta}^{K_1}$, $g_{h,\theta}^{K_2}$, $g_{h,\theta}^{K_3}$ such that $g_{h,\theta} = \sum_{i=0}^{3} g_{h,\theta}^{K_i}$, and

$$\begin{aligned}
&g_{h,\theta}^{K_j}(x) \neq 0 \Longrightarrow g_{h,\theta}^{K_i}(x) = 0, \quad \forall x, \ \forall i \neq j,\\
&g_{h,\theta}^{K_0}(x) = g_{h,\theta}(x), \quad \forall x \in \{\mathbf{0} + (2n, 2m), n, m \in \mathbb{Z}\},\\
&g_{h,\theta}^{K_1}(x) = g_{h,\theta}(x), \quad \forall x = \{\mathbf{0} + (2n, 2m+1), n, m \in \mathbb{Z}\},\\
&g_{h,\theta}^{K_2}(x) = g_{h,\theta}(x), \quad \forall x = \{\mathbf{0} + (2n+1, 2m), n, m \in \mathbb{Z}\},\\
&g_{h,\theta}^{K_3}(x) = g_{h,\theta}(x), \quad \forall x = \{\mathbf{0} + (2n+1, 2m+1), n, m \in \mathbb{Z}\},
\end{aligned}$$

where $\mathbf{0}$ stands for the coordinates of the origin of the kernel. The kernels $g_{h,\theta}^{K_i}$ can be defined also as $g_{h,\theta}^{K_i} = \uparrow_{K_i} \{\downarrow_{K_i} \{g_{h,\theta}\}\}$.

For example, to obtain *denoised* estimates for the G_1 subcomponent at G_1 locations on a directional horizontal or vertical 1D "linewise" window, the cross-color kernel $g_{h,\theta}$ is composed from two kernels, one for function estimation and another for derivative estimation: $g_{h,\theta}^{K_0} = g_{h,\theta}^{(0)}$, $g_{h,\theta}^{K_1} = \alpha g_{h,\theta}^{(1)}$, and the superscripts (0) or (1) denote the order of estimated derivative [4]. The parameter α is a weight of the derivative estimates. These two kernels, when used within the convolution are, respectively, non-zero on the G_1 pixels and non-zero on the R pixels. To obtain *interpolated* estimates for the R component at G_1 locations, the cross-color kernel $g_{h,\theta}$ is instead composed as $g_{h,\theta}^{K_0} = \alpha g_{h,\theta}^{(1)}$, $g_{h,\theta}^{K_1} = g_{h,\theta}^{(0)}$, i.e., by a derivative estimation kernel which (in the convolution) is non-zero on the G_1 pixels and a function estimation kernel which (in the convolution) is non zero on the R pixels. Naturally, the same convolutions produce also denoised R estimates at R and interpolated G at R.

4.2 Variance computation

For the proper deblurring and interpolation it is crucial to calculate correctly the variances of the LPA cross-color estimates. These variances are used by the ICI to enable spatial adaptivity with respect to the signal and the noise.

Let us observe that the simple formula (convolution of the squared kernel against the variances of the Bayer samples) used

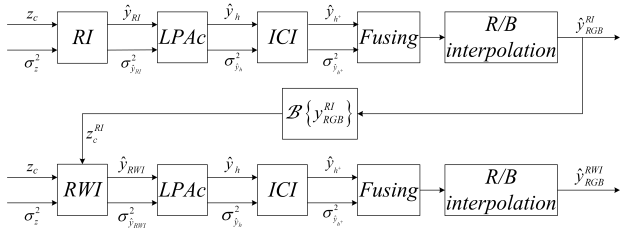

Figure 1: Scheme for joint LPA-ICI based deblurring and CFAI. LPAc denotes cross-color LPA.

for the basic interpolation algorithm [4] for blur-free Bayer data is really inadequate for this deblurring case, as it totally ignores the correlating (coloring) effect of the regularized inverse operator (7).

First, let us perform some manipulations on the convolution (8), aiming at separating the original blurred-Bayer observations (where the noise is independent) from the RI and LPA estimates, where the noise is correlated. To accomplish this, we exploit the downsampling and upsampling operators defined in Section 3.1.

Without loss of generality, let us consider the estimation at G_1 positions. Analogous formulas apply to other components. For these points, the estimate formula (8) can be rewritten as follows:

$$\downarrow_{G_1}\left\{\hat{y}^{\mathrm{RI}}_{h,\theta}\right\} = \sum_{i=0}^{3} z_{c_i} \circledast \left(\varphi^{\mathrm{RI}}_{\mathrm{sub}} \circledast \downarrow_{K_i}\left\{g^{K_i}_{h,\theta}\right\}\right), \qquad (9)$$

where $c_0 = G_1$, $c_1 = R$, $c_2 = B$, $c_3 = G_2$.

The estimate (9) is calculated by a sum of four independent terms, each of which is obtained as a convolution of data with independent noise against a special kernel. Thus, the variances of the estimates (9) are calculated as

$$\downarrow_{G_1}\left\{\sigma^2_{\hat{y}_{h,\theta}}\right\} = \sum_{i=0}^{3} \downarrow_{c_i}\left\{\sigma^2\right\} \circledast \left(\varphi^{\mathrm{RI}}_{\mathrm{sub}} \circledast \downarrow_{K_i}\left\{g^{K_i}_{h,\theta}\right\}\right)^2.$$

It is convenient to compute the convolution against the variance in the frequency domain, because both terms have support size $\frac{N}{2}\times\frac{M}{2}$:

$$\downarrow_{G_1}\left\{\sigma^2_{\hat{y}_{h,\theta}}\right\} = \qquad (10)$$

$$= \sum_{i=0}^{3} \mathcal{F}^{-1}_{\mathrm{sub}}\left(\mathcal{F}_{\mathrm{sub}}\left(\downarrow_{c_i}\left\{\sigma^2\right\}\right) \mathcal{F}_{\mathrm{sub}}\left(\left(\varphi^{\mathrm{RI}}_{\mathrm{sub}} \circledast \downarrow_{K_i}\left\{g^{K_i}_{h,\theta}\right\}\right)^2\right)\right).$$

In particular, if we use "linewise" 1D directional kernels [4], only two instead of four kernels are needed and, as an example, for the estimation at the G_1 components, (9) is reduced to

$$\downarrow_{G_1}\left\{\hat{y}^{\mathrm{RI}}_{h,\theta}\right\} =$$

$$= z_{G_1} \circledast \left(\varphi^{\mathrm{RI}}_{\mathrm{sub}} \circledast \downarrow_{K_0}\left\{g^{(0)}_{h,\theta}\right\}\right) + z_R \circledast \left(\varphi^{\mathrm{RI}}_{\mathrm{sub}} \circledast \downarrow_{K_1}\left\{\alpha g^{(1)}_{h,\theta}\right\}\right).$$

The variances of these estimates are calculated as

$$\downarrow_{G_1}\left\{\sigma^2_{\hat{y}_{h,\theta}}\right\} = \downarrow_{G_1}\left\{\sigma^2\right\} \circledast \left(\varphi^{\mathrm{RI}}_{\mathrm{sub}} \circledast \downarrow_{K_0}\left\{g^{(0)}_{h,\theta}\right\}\right)^2 +$$

$$+ \downarrow_{R}\left\{\sigma^2\right\} \circledast \left(\varphi^{\mathrm{RI}}_{\mathrm{sub}} \circledast \downarrow_{K_1}\left\{\alpha g^{(1)}_{h,\theta}\right\}\right)^2.$$

Similar to equation (10),

$$\downarrow_{G_1}\left\{\sigma^2_{\hat{y}_{h,\theta}}\right\} =$$

$$= \mathcal{F}^{-1}_{\mathrm{sub}}\left(\mathcal{F}_{\mathrm{sub}}\left(\downarrow_{G_1}\left\{\sigma^2\right\}\right) \mathcal{F}_{\mathrm{sub}}\left(\left(\varphi^{\mathrm{RI}}_{\mathrm{sub}} \circledast \downarrow_{K_0}\left\{g^{(0)}_{h,\theta}\right\}\right)^2\right)\right) +$$

$$+ \mathcal{F}^{-1}_{\mathrm{sub}}\left(\mathcal{F}_{\mathrm{sub}}\left(\downarrow_{R}\left\{\sigma^2\right\}\right) \mathcal{F}_{\mathrm{sub}}\left(\left(\varphi^{\mathrm{RI}}_{\mathrm{sub}} \circledast \downarrow_{K_1}\left\{\alpha g^{(1)}_{h,\theta}\right\}\right)^2\right)\right),$$

since all the terms have large supports of size $\frac{N}{2}\times\frac{M}{2}$ and it is more efficient to perform these operations in the frequency domain.

Exploiting the obtained directional estimates $\hat{y}^{\mathrm{RI}}_{h,\theta}(x)$ and their variances $\sigma^2_{\hat{y}_{h,\theta}}$, the CFAI is applied as it is proposed in [4]. The obtained result is a deblurred and interpolated full-size RGB image $\hat{y}^{\mathrm{RI}}_{RGB}$. This estimate is used as a pilot estimate for Wiener filtering.

		L+L(i)	Tr+L(i)	LPAc
14 	Red	30.19	**30.40**	30.24
	Green	30.68	**30.91**	30.83
	Blue	30.45	**30.68**	30.46
19 	Red	29.77	**29.98**	29.87
	Green	29.77	**29.95**	29.87
	Blue	29.83	**29.99**	29.82
23 	Red	35.96	**36.70**	36.06
	Green	36.26	**36.97**	36.43
	Blue	36.09	**36.73**	36.06
Average PSNR over 24 test images	Red	31.45	**31.73**	31.58
	Green	31.67	**31.92**	31.82
	Blue	31.47	**31.70**	31.54

Table 1: PSNR comparison. L+L(i): independent LPA-ICI deblurring [7] of 4 subsampled blurred color components + LPA-ICI CFAI [9]. Tr+L(i): independent transform-based deblurring [8] of 4 sampled blurred color components + LPA-ICI CFAI [9]. LPAc: proposed technique.

4.3 Regularized Wiener inverse

The regularized Wiener inverse is applied as follows:

$$\hat{Y}^{\mathrm{RWI}}_{c} = \frac{V^{*}_{\mathrm{sub}}\left|Z^{\mathrm{RI}}_{c}\right|^2}{\left|V_{\mathrm{sub}}Z^{\mathrm{RI}}_{c}\right|^2 + \sigma^2\varepsilon^2_{\mathrm{RWI}}} Z_c, \qquad c = R, G_1, G_2, B,$$

where $Z^{\mathrm{RI}}_{c} = \mathcal{F}_{\mathrm{sub}}\{z^{\mathrm{RI}}_{c}\}$, Z^{RI}_{c} corresponds to Fourier transform of the sampled estimated colors $z^{\mathrm{RI}}_{c} = \downarrow_{c}\left\{z^{\mathrm{RI}}_{\mathrm{bayer}}\right\}$, $z^{\mathrm{RI}}_{\mathrm{bayer}} = \mathcal{B}\left\{\hat{y}^{\mathrm{RI}}_{RGB}\right\}$ and $\varepsilon^2_{\mathrm{RI}}$ is a regularization parameter. Then, the LPA-ICI CFAI [4] is applied to Bayer-patterned data

$$\hat{y}^{\mathrm{RWI}}_{\mathrm{bayer}} = \uparrow_{R}\left\{\hat{y}^{\mathrm{RWI}}_{R}\right\} + \uparrow_{G_1}\left\{\hat{y}^{\mathrm{RWI}}_{G_1}\right\} + \uparrow_{G_2}\left\{\hat{y}^{\mathrm{RWI}}_{G_2}\right\} + \uparrow_{B}\left\{\hat{y}^{\mathrm{RWI}}_{B}\right\},$$

where $\hat{y}^{\mathrm{RWI}}_{c} = \mathcal{F}^{-1}_{\mathrm{sub}}\left(\hat{Y}^{\mathrm{RWI}}_{c}\right)$. The variance of the directional interpolation and denoising estimates $\hat{y}^{\mathrm{RWI}}_{h,\theta}$ is calculated, without loss of generality for $x \in X_{G_1}$, as

$$\mathrm{var}\left\{\hat{y}^{\mathrm{RWI}}_{h,\theta}(x)\right\} = \left(\sigma^2 \circledast \left(\uparrow_{G_1}\left\{\varphi^{\mathrm{RWI}}_{G_1,\mathrm{sub}} \circledast \downarrow_{G_1}\left\{g^{(0)}_{h,\theta}\right\}\right\} + \right.\right.$$

$$\left.\left. + \uparrow_{R}\left\{\varphi^{\mathrm{RWI}}_{R,\mathrm{sub}} \circledast \downarrow_{R}\left\{g^{(1)}_{h,\theta}\right\}\right\}\right)^2\right)(x) = \sigma^2_{\hat{y}_{h,\theta}}(x),$$

where $\varphi^{\mathrm{RWI}}_{G_1,\mathrm{sub}} = \mathcal{F}^{-1}_{\mathrm{sub}}\left(\frac{V^{*}_{\mathrm{sub}}\left|Z^{\mathrm{RI}}_{G_1}\right|^2}{\left|V_{\mathrm{sub}}Z^{\mathrm{RI}}_{G_1}\right|^2 + \sigma^2\varepsilon^2_{\mathrm{RWI}}}\right)$, is a transform operator for the G_1 subcomponent, and similarly $\varphi^{\mathrm{RWI}}_{R,\mathrm{sub}} = \mathcal{F}^{-1}_{\mathrm{sub}}\left(\frac{V^{*}_{\mathrm{sub}}\left|Z^{\mathrm{RI}}_{R}\right|^2}{\left|V_{\mathrm{sub}}Z^{\mathrm{RI}}_{R}\right|^2 + \sigma^2\varepsilon^2_{\mathrm{RWI}}}\right)$ is a transform operator for the R subcomponent. The obtained result is a deblurred and interpolated full-size RGB image $\hat{y}^{\mathrm{RWI}}_{RGB}$. This is our final estimate.

5. EXPERIMENTS

For the experiments we used the standard Kodak set of 24 color test images of size 768×512 with the intensities normalized in the range [0,1]. The blurred observations are generated using a separable PSF υ=[1,4,6,4,1]$^{\mathrm{T}}$[1,4,6,4,1]/256 and the Poissonian noise model (3) with χ=17600. The use of ICI requires the knowledge of σ in (1); similar to [4], we estimate σ from z_{bayer} as $\hat{\sigma} = \sqrt{z_{\mathrm{bayer}}/\chi}$.

Since the research on deblurring of noisy Bayer data is very recent, it is not possible to carry out extensive and fair comparisons with other techniques and implementations. In particular, in [1] no experiments are shown, [2] presents experiments exclusively

	L+L(i)	Tr+L(i)	LPAc
14	1.8133	**1.6474**	1.7301
19	1.6540	**1.4541**	1.5034
23	1.1948	**0.9338**	1.0673
Average S-CIELAB over 24 test images	1.5130	**1.3357**	1.3933

Table 2: Comparison in terms of S-CIELAB. L+L(i): independent LPA-ICI deblurring [7] of 4 subsampled blurred color components + LPA-ICI CFAI [9]. Tr+L(i): independent transform-based deblurring [8] of 4 sampled blurred color components + LPA-ICI CFAI [9]. LPAc: proposed technique.

for the multiframe case, and [3] considers noise in the theoretical modeling, but not in the experimental part, where only noise-free signals are deblurred. Therefore, we resort to comparing the proposed joint procedure, denoted as "LPAc", versus the state-of-the-art deblurring and interpolation performed as two independent steps, where the interpolation assumes that the deblurred data is noise free. Specifically, for this comparison we consider the independent combination of the LPA-ICI deblurring [7] and LPA-ICI CFAI [9], denoted as "L+L(i)", and the independent combination of the adaptive transform-based deblurring for Poissonian data [8] and LPA-ICI based CFAI [9], denoted as "Tr+L(i)".

The results in terms of PSNR and S-CIELAB values[2] are shown in Tables 1 and 2 for three images from the test set. The average obtained values over all 24 images in the Kodak set are also given. The best results are highlighted with bold face. The average time for processing a 768×512 image with "L+L(i)", "Tr+L(i)", and "LPAc" are 7.43, 25.41, and 4.63 minutes, respectively[3]. From the two tables it can be seen that, despite a much lower computational complexity, the proposed technique is effective and yields results that are competitive to the other methods in the comparison. It is worth noting that even though the results of "LPAc" are slightly inferior to those of "Tr+L(i)" (on average about 0.15 dB difference), the advantage in terms complexity are significant (the proposed algorithm is more than five times faster). We note also that both the deblurring [8] and the interpolation [9] are some of the best to our knowledge.

An illustration of the performance of the proposed joint deblurring-interpolation algorithm is given in Figure 2, where we show a fragment of the noise-free full-size RGB blurred image, of the noisy Bayer-downsampled observations, and of the image restored from these observations using the "LPAc".

6. CONCLUSIONS

The problem of reconstruction of a full-size, full-color, sharp image from the blurred Poissonian noisy Bayer data, simulating image acquisition and reconstruction for single-chip (CCD or CMOS) digital cameras, was considered in this paper. A novel technique for joint deblurring and demosaicing was proposed. The technique is based on the signal-adaptive cross-color filtering [4] and on a two-stage regularized deconvolution [5], specifically designed for Poissonian data. Its efficiency and applicability was shown by numerical and visual quality evaluations comparing to non joint deblurring and demosaicing.

REFERENCES

[1] Trussell, H.J., and R.E. Hartwig,"Mathematics for Demosaicking," *IEEE Trans. Image Process.*, vol. 11, no. 4, pp. 482-492, 2002.

[2]Both PSNR and S-CIELAB were calculated after removing borders of width 15 pixels from the full-size images. This is done in order to avoid influence of the boundary effect on the criteria.

[3]All simulations were performed in the MATLAB environment (ver. 7.4) on a PC equipped with a Pentium 4 HT 3.2GHz CPU and 2 GB of RAM, and running the Windows XP SP2 operating system.

Figure 2: Restoration of the Kodak image no.14. Detail of the blurred RGB image (top-left), of the noisy mosaiced blurred observations (top-right), and of the estimate obtained from these observations using the proposed joint LPA-ICI based deblurring-interpolation "LPAc" (bottom).

[2] Farsiu, S., M. Elad, and P. Milanfar, "Multi-frame demosaicing and super-resolution of color images," *IEEE Trans. Image Process.*, vol. 15, no. 1, pp. 141-159, 2006.

[3] Vega, M., R. Molina, and A.K. Katsaggelos, "A Bayesian Super-Resolution Approach to Demosaicing of Blurred Images," *EURASIP J. Appl. Signal Process.*, vol. 2006, pp. 1-12, 2006.

[4] Paliy, D., A. Foi, R. Bilcu, V. Katkovnik, "Denoising and Interpolation of Noisy Bayer Data with Adaptive Cross-Color Filters", *SPIE-IS&T Electronic Imaging, Visual Comm. Image Process. 2008*, vol. 6822, San Jose, CA, Jan. 2008.

[5] Katkovnik, V., Egiazarian K. and Astola J., "A spatially adaptive nonparametric image deblurring," *IEEE Trans. Image Process.*, vol. 14, no. 10, pp. 1469-1478, Oct. 2005.

[6] Bayer, B.E., "Color imaging array," U.S. Patent 3 971 065, July 1976.

[7] Foi, A., S. Alenius, M. Trimeche, V. Katkovnik, and K. Egiazarian, "A spatially adaptive Poissonian image deblurring", *Proc. IEEE 2005 Int. Conf. Image Process., ICIP 2005*, Genova, Sept. 2005.

[8] Foi, A., S. Alenius, M. Trimeche, and V. Katkovnik, "Adaptive-size block transforms for Poissonian image deblurring", *Proc. Int. TICSP Workshop Spectral Meth. Multirate Signal Process., SMMSP 2006*, Florence, Sept. 2006.

[9] Paliy, D., V. Katkovnik, R. Bilcu, S. Alenius, K. Egiazarian, "Spatially Adaptive Color Filter Array Interpolation for Noiseless and Noisy Data", *Int. J. Imaging Sys. Tech., Sp. Iss. Appl. Color Image Process.*, vol. 17, no. 3, pp. 105-122, Oct. 2007.