Spatially Adaptive Color Filter Array Interpolation for Noiseless and Noisy Data

Dmitriy Paliy,[1] Vladimir Katkovnik,[1] Radu Bilcu,[2] Sakari Alenius,[2] Karen Egiazarian[1]

[1] Institute of Signal Processing, Tampere University of Technology, P.O. Box 553, 33101 Tampere, Finland

[2] Nokia Research Center, Tampere, Finland

Received 25 January 2007; accepted 29 August 2007

ABSTRACT: Conventional single-chip digital cameras use color filter arrays (CFA) to sample different spectral components. Demosaicing algorithms interpolate these data to complete red, green, and blue values for each image pixel, to produce an RGB image. In this article, we propose a novel demosaicing algorithm for the Bayer CFA. For the algorithm design, we assume that, following the concept proposed in (Zhang and Wu, IEEE Trans Image Process 14 (2005), 2167–2178), the initial interpolation estimates of color channels contain two additive components: the true values of color intensities and the errors that are considered as an additive noise. A specially designed signal-adaptive filter is used to remove this so-called demosaicing noise. This filter is based on the local polynomial approximation (LPA) and the paradigm of the intersection of confidence intervals applied to select varying scales of LPA. This technique is nonlinear and spatially-adaptive with respect to the smoothness and irregularities of the image. The presented CFA interpolation (CFAI) technique takes significant advantage from assuming that the original data is noise-free. Nevertheless, in many applications, the observed data is noisy, where the noise is treated as an important intrinsic degradation of the data. We develop an adaptation of the proposed CFAI for noisy data, integrating the denoising and CFAI into a single procedure. It is assumed that the data is given according to the Bayer pattern and corrupted by signal-dependant noise common for charge-coupled device and complementary-symmetry/metal-oxide semiconductor sensors. The efficiency of the proposed approach is demonstrated by experimental results with simulated and real data. © 2007 Wiley Periodicals, Inc. Int J Imaging Syst Technol, 17, 105–122, 2007; Published online in Wiley InterScience (www.interscience.wiley.com). DOI 10.1002/ima.20109

Key words: Bayer pattern; color filter array interpolation; spatially adaptive denoising; sensor noise

I. INTRODUCTION

The common approach in single-chip digital cameras is to use a color filter arrays (CFA) to sample different spectral components like red, green, and blue. The sensor records one value per pixel location. The resulting image is a gray-scale mosaic-like one. Demosaicing algorithm interpolates sets of complete red, green, and blue values for each pixel, to make an RGB image. Independent interpolation of color channels usually leads to drastic color distortions. The way to effectively produce a joint color interpolation plays a crucial role for demosaicing.

Modern efficient algorithms exploit several main facts. The first is the high correlation between the red, green, and blue channels for natural images. As a result, all three color channels are very likely to have the same texture and edge locations. The second fact is that digital cameras use the CFA in which the luminance (green) channel is sampled at the higher rate than the chrominance (red and blue) channels. Therefore, the green channel is less likely to be aliased, and details are preserved better in the green channel than in the red and blue channels (Gunturk et al., 2002). Also, the CFA is a crucial element in design of single-sensor digital cameras. Different characteristics in design of CFA affect both performance and computational efficiency of the demosaicking solution (Adams et al., 1998; Lukac, 2005a). The fundamentals about digital color image acquisition with single-sensor can be found in (Parulski and Spaulding, 2002; Lukac and Plataniotis, 2006b).

Considering the fact that the Bayer CFA (Bayer, 1976) (see Fig. 1) is one of the most often exploited today, in this article, we focus on techniques for this particular CFA.

A. Correlation Models.

There are two basic interplane correlation models: the color difference rule (Laroche and Prescott, 1994; Hamilton and Adams, 1997) and the color ratio rule (Kimmel, 1999; Lukac et al., 2004a). The first model asserts that intensity differences between red, green, and blue channels are slowly varying, that is the differences between color channels are locally nearly-constant (Laroche and Prescott, 1994; Hamilton and Adams, 1997; Adams, 1998; Lukac and Plataniotis, 2004a, 2005b; Hirakawa and Parks, 2005a; Li, 2005; Zhang and Wu, 2005). Thus, they contain low-frequency components only, making the interpolation using the color differences easier (Hirakawa and Parks, 2005a).

Correspondence to: Dmitriy Paliy; e-mail: dmitriy.paliy@tut.fi

This work was supported by the Finnish Funding Agency for Technology and Innovation (Tekes), AVIPA Project.

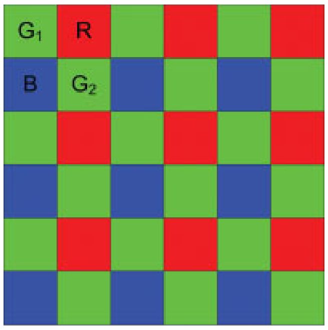


Figure 1. Bayer color filter array. [Color figure can be viewed in the online issue, which is available at www.interscience.wiley.com.]

The second correlation model is based on the assumption that the ratios between colors are constant over some local regions (Kimmel, 1999; Lukac et al., 2004a). This hypothesis follows from the Lambert's law that if two colors have equal chrominance then the ratios between the intensities of three color components are equal (Kimmel, 1999; Hirakawa and Parks, 2005a).

The first difference-based correlation model is found to be more efficient than the ratio-based model and, therefore, exploited more often in practice. Moreover, the color-difference rule can be implemented with a lower computational cost and better fits linear interpolation modeling (Li, 2005).

Other correlation models used in the demosaicing literature can be found in (Keren and Osadchy, 1999; Lukac and Plataniotis, 2004b, 2006a).

B. Demosaicing Methods. Many demosaicing algorithms (Laroche and Prescott, 1994; Hamilton and Adams, 1997; Kimmel, 1999; Lukac et al., 2004) incorporate edge directionality in interpolation. Interpolation along object boundaries is preferable versus interpolation across these boundaries for most of the models.

We will classify the demosaicing techniques into two categories: noniterative (Laroche and Prescott, 1994; Hamilton and Adams, 1997; Adams, 1998; Taubman, 2000; Pei and Tam, 2003; Malvar et al., 2004; Zhang and Wu, 2005; Menon et al., 2007) and iterative (Kimmel, 1999; Gunturk et al., 2002; Lukac et al., 2004a; Hirakawa and Parks, 2005a; Li, 2005). There are also alternative ways of this classification, for instance considered in (Gunturk et al., 2005).

Noniterative demosaicing techniques basically rely on the idea of edge-directed interpolation. In a variety of color demosaicing techniques, the gradient estimates analysis plays a central role in reconstructing sharp edges. The exploitation of intraplane correlation typically is done by estimating local gradients under the main assumption that locally the difference between colors is nearly constant. Directional filtering is the most popular approach for color demosaicing that produces competitive results. The best known directional interpolation scheme is perhaps the method proposed by Hamilton and Adams (1997). The authors use the gradients of blue and red channels as the correction terms to interpolate the green channel. Similar idea is exploited effectively in (Laroche and Prescott, 1994; Wu and Zhang, 2004; Malvar et al., 2004) but with a different aggregation of vertical and horizontal estimates. Directional filtering with a posteriori decision is also effectively exploited for demosaicing in (Menon et al., 2007). A novel efficient data adaptive filtering concept in conjunction with the refined spectral models is proposed in (Lukac and Plataniotis, 2005a) for demosaicing.

In addition, there are approaches based on the following: pattern recognition (Cok, 1994), restoration algorithms (Taubman, 2000; Trussel et al., 2005), sampling theory (Adams, 1998) [see (Gunturk et al., 2005) for more details], the regularization theory (Keren and Osadchy, 1999), the Bayesian approach (Brainard, 1994; Vega et al., 2005), demosaicing in frequency domain (Alleysson et al., 2005; Dubois, 2005). Taubman (2000) proposed an efficient preconditioned approach of Bayesian demosaicing that is used in some digital cameras today.

A number of iterative demosaicing techniques have been proposed recently (Kimmel, 1999; Gunturk et al., 2002; Lukac et al., 2004a; Hirakawa and Parks, 2005a; Li, 2005). It has been observed that iterative demosaicing techniques often demonstrate higher quality restoration than noniterative ones at the price of increased computational cost. The refinement of green pixels and red/blue pixels are mutually dependent and jointly beneficial to each other. An iterative strategy is exploited in (Li, 2005) to handle this correlation.

A new idea has been proposed and effectively used in the recent papers (Li, 2005; Zhang and Wu, 2005), where the differences between initial directional interpolated estimates of color intensities are filtered. In the paper (Zhang and Wu, 2005), the concept of directional ''demosaicing noise'' was introduced for the interpolation errors. A filtering procedure is exploited to remove this noise and obtain the improved estimates of the differences between the chrominance and luminance channels as result of denoising procedure.

Our article is motivated by the work (Zhang and Wu, 2005), where the demosaicing is reformulated as the denoising problem. The differences between color channels are considered as noisy signals, and the term noise is used for interpolation errors. We use a directional anisotropic scale-adaptive denoising technique to remove the errors, instead of the fix-length filter used in (Zhang and Wu, 2005). The exploited technique is based on the local polynomial approximation (LPA). The adaptivity to data is provided by the multiple hypothesis testing called the intersection of confidence intervals (ICI) rule, which is applied to select varying scales (window sizes) of LPA (Katkovnik, 1999; Katkovnik et al., 2002, 2006). The main problem that appears is that the LPA-ICI requires a priori knowledge about the variance of the noise. However, the ''demosaicing noise'' cannot be considered as a stationary one and its statistics are unknown, since the errors strongly depend on the signal. For instance, its variance near edges may be significantly higher than at smooth areas and, therefore, in this article, it is estimated locally. In such a way, we aim to improve results with a filtering that is adaptive to image irregularities, e.g., edges.

C. Demosaicing of Noisy Data. In many applications, the observed data is noisy. The light passes through the optical system of the camera and is focused on a digital sensor. The sensor is composed of photon-collection sites. Each site works as a photon-counter to measure the amount of light coming to it. The sensor produces a digital value for each site, which corresponds to the intensity of the light at that position. This digital output of the sensor is called ''raw data.''

It is known that the raw data from the sensor is corrupted by signal-dependant noise (Hirakawa and Parks, 2005b; Foi et al., 2006, 2007). Such modeling is common for charge-coupled device (CCD) and complementary-symmetry/metal-oxide semiconductor (CMOS) digital image sensors. A property of such a noise is the dependence of its variance from the signal (Foi, 2007). The most widely encountered models for this dependence are film-grain, multiplicative, speckle noise, and, in particular, Poissonian noise.

The problem is to restore the true observation scene from the noisy subsampled data. The conventional approach used in image restoration chains for raw sensor data exploits successive independent denoising and demosaicing steps. Denoising aims to remove the noise, and demosaicing performs interpolation of missing colors assuming that the processed data is noiseless.

In case of treating the original noisy observed data, the denoising done first was proven to be more efficient. Some post-CFA interpolation (post-CFAI) and pre-CFAI denoising techniques are compared in (Kalevo and Rantanen, 2002). The authors show the possibility to reduce more noise with the pre-CFAI denoising than with the post-CFAI denoising. Also, the computational costs can be lower with the pre-CFAI denoising than with the post-CFAI one. The model of noise plays a crucial role in image denoising, which is known before CFAI but not after CFAI.

Noting that image interpolation and image denoising are both estimation problems, the papers (Hirakawa and Parks, 2005b, 2006) propose a unified approach to performing demosaicing and image denoising jointly, where the noise is modeled as multiplicative Gaussian. The multicolored demosaicing/denoising problem was simplified to a single-color denoising problem. The authors verified that performing demosaicing and denoising jointly is more effective than treating them independently (Hirakawa and Parks, 2005b). Ramanath and Snyder (2003) proposed a bilateral filtering-based scheme to denoise, sharpen, and demosaic the image simultaneously.

Most denoising techniques are designed for stationary Gaussian-distributed noise. We propose a technique specially designed for filtering not only Gaussian but also more general signal-dependant noise. It is natural to adopt the CFAI proposed for noiseless case to the noisy one, since it already considers demosaicing as a denoising problem. The crucial difference is that here all pixels have to be either denoised or interpolated from noisy observations. Therefore, filtering only the difference between directional interpolated estimates of color intensities is not sufficient, and thus we decorrelate them by calculating both sum and difference, and then apply the LPA-ICI denoising to these pairs of components. The advantage of this approach is that different color channels are used for both denoising and interpolation, filtering also ''demosaicing noise'' that is present implicitly on the image. The proposed technique results in better utilization of data, in better performance and quality of image restoration, and lower complexity of implementation. These issues are of crucial importance especially for small mobile devices, where the impact of noise is particularly severe because of the constrained power and hardware.

Preliminary results obtained in (Paliy et al., 2007a,b) show the efficiency of this approach.

The structure of the article is the following. In Section II, we propose a method of CFAI for noiseless data. In Section III, we describe the technique's adaptation for the noisy Bayer data. Finally, the simulation results are shown in Section IV.

II. COLOR FILTER ARRAY INTERPOLATION OF NOISELESS DATA

A. Image Formation Model for Noiseless Data. We follow the general Bayer mask image formation model:

$$z_{\text{bayer}}(i,j) = \Psi\{y_{RGB}(i,j)\}, \tag{1}$$

where $\Psi\{\cdot\}$ is a Bayer sampling operator (Bayer, 1976)

$$\Psi\{y_{RGB}(i,j)\} = \begin{cases} G(i,j), & \text{at } (i,j) \in X_{G_1}, \\ G(i,j), & \text{at } (i,j) \in X_{G_2}, \\ R(i,j), & \text{at } (i,j) \in X_R, \\ B(i,j), & \text{at } (i,j) \in X_B, \end{cases} \tag{2}$$

and z_{bayer} is an output signal of the sensor, $y_{RGB}(i,j) = (R(i,j), G(i,j), B(i,j))$ is a true color RGB observation scene, $X = \{(i,j)\colon i = 1, \ldots, 2N, j = 1, \ldots, 2M\}$ are the spatial coordinates, and R (red), G (green), and B (blue) correspond to the color channels. For two available green channels, we will use notations $G_1(i,j)$, such that $(i,j) \in X_{G_1} = \{(i,j)\colon i = 1, 3, \ldots, 2N-1, j = 1, 3, \ldots, 2M-1\}$, and $G_2(i,j)$, such that $(i,j) \in X_{G_2} = \{(i,j)\colon i = 2, 4, \ldots, 2N, j = 2, 4, \ldots, 2M\}$. Spatial coordinates for the red $R(i,j)$ and blue $B(i,j)$ color channels are denoted $X_R = \{(i,j)\colon i = 1, 3, \ldots, 2N-1, j = 2, 4, \ldots, 2M\}$ and $X_B = \{(i,j)\colon i = 2, 4, \ldots, 2N, j = 1, 3, \ldots, 2M-1\}$, respectively.

Demosaicing attempts to invert $\Psi\{\cdot\}$ to reconstruct $R(i,j)$, $G(i,j)$, and $B(i,j)$ intensities from the observations $z_{\text{bayer}}(i,j)$.

As in (Zhang and Wu, 2005), our algorithm consists of the following steps: initialization, filtering, and interpolation. At the initialization, the approximate color estimates are obtained and directional differences between G–R and G–B are calculated. These differences are considered as degraded by noise and filtered. The modified version of the LPA-ICI algorithm is used for this filtering. Finally, the obtained estimates are exploited to calculate missing color values at each pixel.

B. Initialization. Firstly, we calculate the directional (horizontal and vertical) estimates of the green channel at every point $(i,j) \in X$ following the rules of Hamilton–Adams algorithm (Hamilton and Adams, 1997). Interpolation of G at R positions $(i,j) \in X_R$ is done as follows:

$$\tilde{G}_h(i,j) = \frac{1}{2}(G(i+1,j) + G(i-1,j)) + \frac{1}{4}(-R(i-2,j) + 2R(i,j) - R(i+2,j)), \tag{3}$$

$$\tilde{G}_v(i,j) = \frac{1}{2}(G(i,j+1) + G(i,j-1)) + \frac{1}{4}(-R(i,j-2) + 2R(i,j) - R(i,j+2)). \tag{4}$$

Here, h and v stand for horizontal and vertical estimates. Similar (3) and (4), the initial directional estimates for the red channel R at the green positions G ($(i,j) \in X_{G_1}$ or $(i,j) \in X_{G_2}$) are interpolated as:

$$\tilde{R}_h(i,j) = \frac{1}{2}(R(i+1,j) + R(i-1,j)) + \frac{1}{4}(-G(i-2,j) + 2G(i,j) - G(i+2,j)), \tag{5}$$

$$\tilde{R}_v(i,j) = \frac{1}{2}(R(i,j+1) + R(i,j-1)) + \frac{1}{4}(-G(i,j-2) + 2G(i,j) - G(i,j+2)). \tag{6}$$

As a result of (3)–(4) and (5)–(6), we obtain at the every horizontal line of red and green values two sets of true (from data) and estimated (interpolated) green and red values:

$$\begin{array}{ccccccc} ... & \tilde{G}_h & G & \tilde{G}_h & G & \tilde{G}_h & ... \\ ... & R & \tilde{R}_h & R & \tilde{R}_h & R & ... \end{array}.$$

Similar calculations are produced for the vertical lines.

At every point, the differences between the true values $R(i,j)$ and $G(i,j)$ and the directional estimates $\tilde{G}_h(i,j)$ and $\tilde{R}_h(i,j)$ are calculated as follows:

$$\tilde{\Delta}^h_{g,r}(i,j) = G(i,j) - \tilde{R}_h(i,j), \quad (i,j) \in X_{G_1}, \tag{7}$$

and

$$\tilde{\Delta}^h_{g,r}(i,j) = \hat{G}_h(i,j) - R(i,j), \quad (i,j) \in X_R, \tag{8}$$

for the horizontal direction. For the vertical direction, the analogous computations are:

$$\tilde{\Delta}^v_{g,r}(i,j) = G(i,j) - \tilde{R}_v(i,j), \quad (i,j) \in X_{G_2}, \tag{9}$$

and

$$\tilde{\Delta}^v_{g,r}(i,j) = \hat{G}_v(i,j) - R(i,j), \quad (i,j) \in X_R. \tag{10}$$

As in (Zhang and Wu, 2005), we assume for further filtering that these differences between the intensities of different color channels can be presented as the sums of the true values of the underlying differences and errors:

$$\tilde{\Delta}^h_{g,r}(i,j) = \Delta_{g,r}(i,j) + \varepsilon^h_{g,r}(i,j), \tag{11}$$

$$\tilde{\Delta}^v_{g,r}(i,j) = \Delta_{g,r}(i,j) + \varepsilon^v_{g,r}(i,j), \tag{12}$$

where $\varepsilon_h(i,j)$ and $\varepsilon_v(i,j)$ are considered as random demosaicing noise; $\Delta_{g,r}(i,j)$ is the true difference between green and red color channels.

The blue channel B is treated in the same way, and we calculate the directional differences $\tilde{\Delta}^h_{g,b}(i,j)$ and $\tilde{\Delta}^v_{g,b}(i,j)$.

C. LPA-ICI Filtering of Directional Differences. The LPA-ICI filtering (Katkovnik et al., 2002, 2006) is used for all noisy estimates $\tilde{\Delta}^h_{g,r}(i,j)$, $\tilde{\Delta}^v_{g,r}(i,j)$ for R, and $\tilde{\Delta}^h_{g,b}(i,j)$, $\tilde{\Delta}^v_{g,b}(i,j)$ for B. To introduce this filtering in the form applicable for any input data, we assume for a moment that this input noisy data have the form:

$$z(i,j) = y(i,j) + n(i,j), \tag{13}$$

where $(i,j) \in X$, $z(i,j)$ is a noisy observation, $y(i,j)$ is a true signal and $n(i,j)$ is a noise.

The LPA is a general tool for linear filter design, in particular for design of the directional filters of the given orders on the arguments i and j. Let $g_{s,\theta}$ be the impulse response of the directional linear filter designed by the LPA (Katkovnik et al., 2006), where θ is a direction of smoothing and s is a scale parameter (window size of the filter)*. The details of the filter design are given in Appendix. The exploited linear filter is obtained as a linear combination of two 1D filters:

$$g_{s,\theta}(i,j) = (1-\alpha)g^0_{s,\theta}(i,j) + \alpha g^1_{s,\theta}(i,j), \tag{14}$$

where $g^0_{s,\theta}$ is a zero-order polynomial kernel, $g^1_{s,\theta}$ is a first-order polynomial kernel, and s is the length of the filters. The mixing parameter α is fixed to be 0.1 in this article. Approximation with larger than zero-order often results in higher instability of filtering, while zero-order often results in lower performance. Therefore, we exploit a combination of zero- and first-order kernels.

A set of the image estimates of different scales s and different directions θ are calculated by the convolution

$$\widehat{y}_{s,\theta}(i,j) = (z \circledast g_{s,\theta})(i,j), \tag{15}$$

for $s \in S = \{s_1, s_2, \ldots, s_J\}$, where $s_1 < s_2 < \ldots < s_J$ and $\theta \in \Theta$.

The ICI rule is the algorithm for selection of the scale (close to the optimal least-square value) for every pixel (i,j) (Katkovnik et al., 2006). This algorithm uses a sequence of confidence intervals

$$D_s = \left[\widehat{y}_{s,\theta}(i,j) - \Gamma\sigma_{\hat{y}_{s,\theta}}, \widehat{y}_{s,\theta}(i,j) + \Gamma\sigma_{\hat{y}_{s,\theta}}\right], \quad s \in S, \tag{16}$$

where $\Gamma > 0$ is a threshold parameter for the ICI, $\widehat{y}_{s,\theta}$ is the estimate of y, and $\sigma_{\hat{y}_{s,\theta}}$ is the standard deviation of this estimate.

In this article, the standard deviation in (16) is calculated as the weighted mean of the squared errors between the estimate and the observations in the directional neighborhood of the pixel (i,j):

$$\sigma_{\hat{y}_{s,\theta}}(i,j) = \sqrt{((z - \hat{y}_{s,\theta})^2 \circledast g^2_{s,\theta})(i,j)}, \tag{17}$$

where the weights are defined by $g_{s,\theta}$ used in (15).

Note that in the standard form of the ICI (Katkovnik et al., 2006) the standard deviation of the estimate is different from (17) and calculated as $\sigma_{\hat{y}_{s,\theta}}(i,j) = \sqrt{(\sigma^2 \circledast g^2_{s,\theta})(i,j)}$, where σ is a given standard deviation of the additive signal-independent observation noise in the model (13). However, in the considered models and (11) and (12), we deal with the data where the noise is only a convenient form for modeling of the interpolation errors that are actually nonrandom. Thus, the standard deviation of the estimate $\widehat{y}_{s,\theta}$ is estimated locally by (17) at every position (i,j) over the directional local area.

The ICI rule defines the adaptive scale as the largest s^+ of those scales in S whose estimate does not differ significantly from the estimates corresponding to the smaller window sizes. This optimization of s for each of the directional estimates yields the adaptive scales $s^+(\theta)$ for each direction θ. The union of the supports of $g_{s+(\theta),\theta}$ is considered as an approximation of the best local vicinity of (i,j) in which the estimation model fits the data. The final estimate is calculated as a linear combination of the obtained adaptive directional estimates $\widehat{y}_{s^+,\theta}(i,j)$.

The final LPA-ICI estimate $\hat{y}(i,j)$ combined from the directional ones is computed as the weighted mean

$$\hat{y}(i,j) = \sum\nolimits_{\theta\in\Theta} \hat{y}_{s^+,\theta}(i,j) w_\theta, \quad w_\theta = \frac{\sigma^{-2}_{\hat{y}_{s^+,\theta}}}{\sum_{\theta\in\Theta} \sigma^{-2}_{\hat{y}_{s^+,\theta}}}, \tag{18}$$

[1]*The MATLAB code that implements the LPA-ICI techique is available in the following link: http://www.cs.tut.fi/~lasip/.

with the variance $\sigma_{\hat{y}}^2$ of the final estimate $\hat{y}(i,j)$ computed for simplicity as

$$\sigma_{\hat{y}}^2 = \left(\sum\nolimits_{\theta\in\Theta} \sigma^{-2}_{\hat{y}_{s^+,\theta}}\right)^{-1}. \tag{19}$$

It is convenient to treat this complex LPA-ICI multidirectional algorithm as an adaptive filter with the input z and the output $\hat{y}$. The input–output equation can be written as $\hat{y} = \mathcal{LI}\{z\}$ by denoting the calculations imbedded in this algorithm as an $\mathcal{LI}$ operator.

Applying the ICI in the form (16)–(18) to $\tilde{\Delta}^h_{g,r}(i,j)$ and $\tilde{\Delta}^v_{g,r}(i,j)$, we obtain the corresponding spatially adaptive estimates of these differences. Let us denote these estimates as $\hat{\Delta}^h_{g,r}(i,j) = \mathcal{LI}\left\{\tilde{\Delta}^h_{g,r}(i,j)\right\}$ and $\hat{\Delta}^v_{g,r}(i,j) = \mathcal{LI}\left\{\tilde{\Delta}^v_{g,r}(i,j)\right\}$.

As in (Zhang and Wu, 2005), combining these vertical and horizontal estimates, we arrive to the final estimate $\hat{\Delta}_{g,r}(i,j)$ in the form

$$\hat{\Delta}_{g,r}(i,j) = \frac{\sigma^{-2}_{\hat{\Delta}^h_{g,r}}}{\sigma^{-2}_{\hat{\Delta}^h_{g,r}} + \sigma^{-2}_{\hat{\Delta}^v_{g,r}}}\hat{\Delta}^h_{g,r}(i,j) + \frac{\sigma^{-2}_{\hat{\Delta}^v_{g,r}}}{\sigma^{-2}_{\hat{\Delta}^h_{g,r}} + \sigma^{-2}_{\hat{\Delta}^v_{g,r}}}\hat{\Delta}^v_{g,r}(i,j), \tag{20}$$

where $\sigma_{\hat{\Delta}^h_{g,r}}$ and $\sigma_{\hat{\Delta}^v_{g,r}}$ are the corresponding standard deviations of $\hat{\Delta}^h_{g,r}$ and $\hat{\Delta}^v_{g,r}$ calculated as in (19).

Similar adaptive LPA-ICI filtering is applied to the differences $\tilde{\Delta}^h_{g,b}(i,j)$ and $\tilde{\Delta}^v_{g,b}(i,j)$ between G and B color channels to obtain the estimate $\hat{\Delta}_{g,b}(i,j)$.

D. Interpolation of *G* Color. From (7) to (10), the interpolated green color at R ($(i,j) \in X_R$) and B ($(i,j) \in X_B$) positions are calculated as follows:

$$\hat{G}(i,j) = R(i,j) + \hat{\Delta}_{g,r}(i,j),\ (i,j) \in X_R,$$
$$\hat{G}(i,j) = B(i,j) + \hat{\Delta}_{g,b}(i,j),\ (i,j) \in X_B,$$

where $\hat{\Delta}_{g,r}$ is the estimate of $G - R$, and $\hat{\Delta}_{g,b}$ is the estimate of $G - B$, obtained using the LPA-ICI technique (16)–(20) described in the previous section.

E. Interpolation of *R*/*B* Colors at *B*/*R* Positions. For the interpolation of R/B colors at B/R positions, we propose to use a special shift-invariant interpolation filter giving the estimates by the standard convolution. This filter has been designed using the LPA for the subsampled grid, which corresponds to R/B channel (Fig. 1) with the symmetrical window function, as shown in Appendix. A variety of polynomial orders and support sizes has been tested. Finally, the second-order polynomial interpolation filter g_{rb} has been chosen

$$g_{rb} = \begin{bmatrix} 0 & 0 & -0.0313 & 0 & -0.0313 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -0.0313 & 0 & 0.3125 & 0 & 0.3125 & 0 & -0.0313 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -0.0313 & 0 & 0.3125 & 0 & 0.3125 & 0 & -0.0313 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -0.0313 & 0 & -0.0313 & 0 & 0 \end{bmatrix}. \tag{21}$$

Interpolation with this filter showed better results than lower order bilinear interpolation used in (Zhang and Wu, 2005).

Then, the interpolated estimates are computed as follows:

$$\hat{R}(i,j) = \hat{G}(i,j) - (\hat{\Delta}_{g,r}\circledast g_{rb})(i,j),\ (i,j) \in X_B, \tag{22}$$
$$\hat{B}(i,j) = \hat{G}(i,j) - (\hat{\Delta}_{g,b}\circledast g_{rb})(i,j),\ (i,j) \in X_R, \tag{23}$$

where $\hat{\Delta}_{g,r}$ is the LPA-ICI estimate of $G - R$, and $\hat{\Delta}_{g,b}$ is the LPA-ICI estimate of $G - B$.

F. Interpolation of *R*/*B* Colors at *G* Positions. As in (Zhang and Wu, 2005), for interpolation of R/B colors at G positions ($(i,j) \in X_{G_1} \cup X_{G_2}$), we use the simplest zero-order interpolation kernel g:

$$g = \begin{bmatrix} 0 & 0.25 & 0 \\ 0.25 & 0 & 0.25 \\ 0 & 0.25 & 0 \end{bmatrix}. \tag{24}$$

Our study showed that the higher order interpolation did not provide significant improvement in performance. The interpolated estimates are computed as follows:

$$\hat{R}(i,j) = G(i,j) - ((\hat{\Delta}_{g,r}\circledast g_{rb})\circledast g)(i,j), \tag{25}$$
$$\hat{B}(i,j) = G(i,j) - ((\hat{\Delta}_{g,b}\circledast g_{rb})\circledast g)(i,j). \tag{26}$$

III. ADAPTATION OF COLOR FILTER ARRAY INTERPOLATION TO NOISY DATA

The general Bayer mask image formation model for the data corrupted by noise is considered as follows:

$$z_{\text{bayer}}(i,j) = \Psi\{y_{RGB}(i,j)\} + \sigma_{\text{bayer}}(i,j)n(i,j). \tag{27}$$

where the term $n(i,j)$ is an independent zero-mean noise with variance equal to one at every point (i,j). Thus, $\sigma_{\text{bayer}}(i,j)$ is the standard deviation of $z_{\text{bayer}}(i,j)$. It is not necessarily invariant with respect to the spatial variable (i,j). The problem is to reconstruct the true color image y_{RGB} from the noisy subsampled data z_{bayer}.

A. Initialization. Considering the fact that not only interpolation has to be performed to reconstruct y_{RGB} but also denoising at every point, the initialization differs from the one presented in Section II-A. One way is to exploit a denoising before CFAI and then perform interpolation treating the obtained data as noiseless. This approach is trivial but may be significantly improved. We aim to perform both denoising and interpolation exploiting the high correlation between color channels and consider not only the differences between color channels but also their sums. Hence, the initialization (7)–(10) is transformed in the following way, where knowledge about σ_{bayer} plays crucial role.

Assuming that the color channels are correlated, we decorrelate them using the following summation and differentiation linear operators working in the horizontal direction:

$$\begin{pmatrix} \tilde{\Phi}^h_{g,r}(i,j) \\ \tilde{\Delta}^h_{g,r}(i,j) \end{pmatrix} = \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}\begin{pmatrix} z_{\text{bayer}}(i,j) \\ \tilde{R}_h(i,j) \end{pmatrix},\quad (i,j) \in X_{G_1}, \tag{28}$$

and

$$\begin{pmatrix} \tilde{\Phi}^h_{g,r}(i,j) \\ \tilde{\Delta}^h_{g,r}(i,j) \end{pmatrix} = \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}\begin{pmatrix} \tilde{G}_h(i,j) \\ z_{\text{bayer}}(i,j) \end{pmatrix},\quad (i,j) \in X_R. \tag{29}$$

For the vertical directions, the corresponding $\hat{\Phi}^{v}_{g,r}(i,j)$ and $\hat{\Delta}^{v}_{g,r}(i,j)$ are calculated as follows:

$$\begin{pmatrix} \tilde{\Phi}^{v}_{g,r}(i,j) \\ \tilde{\Delta}^{v}_{g,r}(i,j) \end{pmatrix} = \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \begin{pmatrix} z_{\text{bayer}}(i,j) \\ \tilde{R}_{v}(i,j) \end{pmatrix}, \quad (i,j) \in X_{G_2}, \tag{30}$$

and

$$\begin{pmatrix} \tilde{\Phi}^{v}_{g,r}(i,j) \\ \tilde{\Delta}^{v}_{g,r}(i,j) \end{pmatrix} = \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \begin{pmatrix} \tilde{G}_{v}(i,j) \\ z_{\text{bayer}}(i,j) \end{pmatrix}, \quad (i,j) \in X_{R}. \tag{31}$$

Here, $\tilde{R}_h, \tilde{G}_h, \tilde{R}_v, \tilde{G}_v$ are calculated similar to (3)–(6) for the noisy data (27). Let us stress that in (3)–(6), G and R notations are used because $z_{\text{bayer}}(i,j) = G(i,j)$, $(i,j) \in X_{G_1} \cup X_{G_2}$, and $z_{\text{bayer}}(i,j) = R(i,j)$, $(i,j) \in X_R$.

We assume for further filtering that the directional differences between the green and red signals $\tilde{\Delta}^{h}_{g,r}(i,j)$ and $\tilde{\Delta}^{v}_{g,r}(i,j)$ can be presented as the sums of the true values of these differences and the errors including the random observation noise in (27) and what has been called the ''directional demosaicing noise'' (Zhang and Wu, 2005):

$$\tilde{\Delta}^{h}_{g,r}(i,j) = \Delta^{h}_{g,r}(i,j) + \varepsilon^{\Delta,h}_{g,r}(i,j), \quad (i,j) \in X_R \cup X_{G_1}, \tag{32}$$

$$\tilde{\Delta}^{v}_{g,r}(i,j) = \Delta^{v}_{g,r}(i,j) + \varepsilon^{\Delta,v}_{g,r}(i,j), \quad (i,j) \in X_R \cup X_{G_2}, \tag{33}$$

where $\varepsilon_{g,r}{}^{\Delta,h}(i,j)$ and $\varepsilon_{g,r}{}^{\Delta,v}(i,j)$ are the errors and $\Delta_{g,r}{}^{h}(i,j)$ and $\Delta_{g,r}{}^{v}(i,j)$ are the true values of the corresponding differences.

The same modeling with the additive errors is assumed for the sums $\tilde{\Phi}^{h}_{g,r}(i,j)$ and $\tilde{\Phi}^{v}_{g,r}(i,j)$:

$$\tilde{\Phi}^{h}_{g,r}(i,j) = \Phi^{h}_{g,r}(i,j) + \varepsilon^{\Phi,h}_{g,r}(i,j), \ (i,j) \in X_R \cup X_{G_1}, \tag{34}$$

$$\tilde{\Phi}^{v}_{g,r}(i,j) = \Phi^{v}_{g,r}(i,j) + \varepsilon^{\Phi,v}_{g,r}(i,j), \ (i,j) \in X_R \cup X_{G_2}, \tag{35}$$

where $\Phi_{g,r}{}^{h}(i,j)$ and $\Phi_{g,r}{}^{v}(i,j)$ are the true values of the sums, and $\varepsilon_{g,r}{}^{\Phi,h}(i,j)$ and $\varepsilon_{g,r}{}^{\Phi,h}(i,j)$ are the errors.

It can be verified that (28)–(31) can be computed as a convolution of $z_{\text{bayer}}(i,j)$ with the linear FIR filters $f_{\Phi} = (-1,2,6,2,-1)/4$ and $f_{\Delta} = (-1,2,-2,2,-1)/4$. For calculations of the variance of the sums $\tilde{\Phi}^{h}_{g,r}$ and $\tilde{\Phi}^{v}_{g,r}$, and differences $\tilde{\Delta}^{h}_{g,r}$ and $\tilde{\Delta}^{v}_{g,r}$ in (28)–(31), we assume that the random observation noise *is dominant* in the errors in (32)–(35). Then the observation noise from (27) gives the following standard deviations for the sums $\tilde{\Phi}^{h}_{g,r}$ and $\tilde{\Phi}^{v}_{g,r}$ (34) and (35):

$$\sigma_{\tilde{\Phi}^{h}_{g,r}}(i,j) = \sqrt{\left(\sigma^{2}_{\text{bayer}} \circledast f^{2}_{\Phi}\right)(i,j)}, \quad (i,j) \in X_R \cup X_{G_1}, \tag{36}$$

$$\sigma_{\tilde{\Phi}^{v}_{g,r}(i,j)} = \sqrt{\left(\sigma^{2}_{\text{bayer}} \circledast \left(f^{\mathrm{T}}_{\Phi}\right)^{2}\right)(i,j)}, \quad (i,j) \in X_R \cup X_{G_2}, \tag{37}$$

where the "T" denotes the transpose operation and $\sigma_{\text{bayer}}(i,j)$ is the noise standard deviation in (27). The standard deviations for the differences $\tilde{\Delta}^{h}_{g,r}$ and $\tilde{\Delta}^{v}_{g,r}$ corresponding to the observation noise are computed as

$$\sigma_{\tilde{\Delta}^{h}_{g,r}}(i,j) = \sqrt{\left(\sigma^{2}_{\text{bayer}} \circledast f^{2}_{\Delta}\right)(i,j)}, \quad (i,j) \in X_R \cup X_{G_1}, \tag{38}$$

$$\sigma_{\tilde{\Delta}^{v}_{g,r}(i,j)} = \sqrt{\left(\sigma^{2}_{\text{bayer}} \circledast \left(f^{\mathrm{T}}_{\Delta}\right)^{2}\right)(i,j)}, \quad (i,j) \in X_R \cup X_{G_2}. \tag{39}$$

The blue channel B is treated in the same way to calculate the directional sums and differences $\tilde{\Delta}^{h}_{g,b}$, $\tilde{\Delta}^{v}_{g,b}$, $\tilde{\Phi}^{h}_{g,b}$, and $\tilde{\Phi}^{v}_{g,b}$ for $(G - B)$ and $(G + B)$.

The spatially adaptive LPA-ICI filtering is exploited to denoise $\tilde{\Delta}^{h}_{g,r}$, $\tilde{\Delta}^{v}_{g,r}$, $\tilde{\Phi}^{h}_{g,r}$, and $\tilde{\Phi}^{v}_{g,r}$ for R color channel, and $\tilde{\Delta}^{h}_{g,b}$, $\tilde{\Delta}^{v}_{g,b}$, $\tilde{\Phi}^{h}_{g,b}$, and $\tilde{\Phi}^{v}_{g,b}$ for B color channel. The standard deviation of the estimate $\widehat{y}_{s,\theta}$ is calculated as

$$\sigma_{\hat{y}_{s,\theta}}(i,j) = \sqrt{(\sigma^{2} \circledast g^{2}_{s,\theta})(i,j)}, \tag{40}$$

instead of (17), where $g_{s,\theta}$ is defined in (15) and σ is calculated as in (36)–(39).

Applying the $\mathcal{LI}$ operator to the calculated sums and differences, we obtain the following denoised estimates of these difference: $\hat{\Delta}^{h}_{g,r}$, $\hat{\Delta}^{v}_{g,r}$, $\hat{\Phi}^{h}_{g,r}$, and $\hat{\Phi}^{v}_{g,r}$ for R color channel, and $\hat{\Delta}^{h}_{g,b}$, $\hat{\Delta}^{v}_{g,b}$, $\hat{\Phi}^{h}_{g,b}$, and $\hat{\Phi}^{v}_{g,b}$ for B color channel. When all the LPA-ICI estimates $\hat{\Delta}^{h}_{g,r} = \mathcal{LI}\left\{\tilde{\Delta}^{h}_{g,r}, \sigma_{\tilde{\Delta}^{h}_{g,r}}\right\}$, $\hat{\Phi}^{h}_{g,r} = \mathcal{LI}\left\{\tilde{\Phi}^{h}_{g,r}, \sigma_{\tilde{\Phi}^{h}_{g,r}}\right\}$, etc. are obtained, they can be used to calculate R, G, and B color components at every position (i,j). Here, we use the modified notation for the operator $\mathcal{LI}$, where the second argument shows the standard deviation of the input signal. These standard deviations $\sigma_{\tilde{\Delta}^{h}_{g,r}}$ and $\sigma_{\tilde{\Phi}^{h}_{g,r}}$ are calculated according to (36)–(39).

B. Interpolation of *G* Component at *R*/*B* Positions and Denoising of *R*/*B* at *R*/*B* Positions.

Aggregation of the horizontal and vertical LPA-ICI estimates of the sums and differences in the final estimates of the sums and difference is produced according to the formula

$$\hat{\Delta}_{g,r} = \frac{\sigma^{-2}_{\hat{\Delta}_h}}{\sigma^{-2}_{\hat{\Delta}_h} + \sigma^{-2}_{\hat{\Delta}_v}} \hat{\Delta}^{h}_{g,r} + \frac{\sigma^{-2}_{\hat{\Delta}_v}}{\sigma^{-2}_{\hat{\Delta}_h} + \sigma^{-2}_{\hat{\Delta}_v}} \hat{\Delta}^{v}_{g,r}, \tag{41}$$

where $\hat{\Delta}^{h}_{g,r} = \mathcal{LI}\left\{\tilde{\Delta}^{h}_{g,r}, \sigma_{\tilde{\Delta}^{h}_{g,r}}\right\}$, $\hat{\Delta}^{v}_{g,r} = \mathcal{LI}\left\{\tilde{\Delta}^{v}_{g,r}, \sigma_{\tilde{\Delta}^{v}_{g,r}}\right\}$, and $\sigma_{\hat{\Delta}_h}$ and $\sigma_{\hat{\Delta}_v}$ are the corresponding standard deviations of $\hat{\Delta}^{h}_{g,r}$ and $\hat{\Delta}^{v}_{g,r}$ calculated as in (19). Aggregation of the horizontal and vertical LPA-ICI estimates $\hat{\Phi}^{h}_{g,r}$ and $\hat{\Phi}^{v}_{g,r}$ in the final estimate $\hat{\Phi}_{g,r}$ is performed in analogous way to (41).

When the estimates of the sums and the differences are calculated, we invert (28)–(31) to calculate the signals from these sums and differences. It results in both interpolation of G at R positions and denoising of R performed as follows:

$$\begin{pmatrix} \hat{G}(i,j) \\ \hat{R}(i,j) \end{pmatrix} = \frac{1}{2} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \begin{pmatrix} \hat{\Phi}_{g,r}(i,j) \\ \hat{\Delta}_{g,r}(i,j) \end{pmatrix}, \quad (i,j) \in X_R, \tag{42}$$

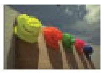
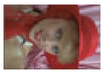
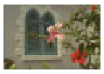
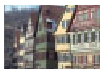
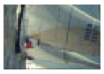
where $\hat{G}$ and $\hat{R}$ are the obtained color estimates.

Similar adaptive LPA-ICI filtering is applied to the B channel with the following reconstruction of the interpolated and denoised signals according to the formula analogous to (42).

C. Denoising of *G* Color.

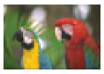
At every point $(i,j) \in X_{G_1}$ or $(i,j) \in X_{G_2}$ we have only vertical or horizontal sums and differences. For instance, it is easy to see from (32) to (35) that at G_1 positions (i,j)

Table I. PSNR comparison for different demosaicing methods computed excluding 15 pixels border.[a] [Color figure can be viewed in the online issue, which is available at www.interscience.wiley.com.]

			HA	LI	HD	SA	DFPD	AP	CCA	CCA+PP	DLMMSE	LPA-ICI	Oracle Γ
01		Red	33.17	30.87	34.51	36.99	36.15	36.69	36.22	37.25	37.58	**39.49**	39.91
		Green	34.65	35.61	36.13	40.76	38.14	40.42	39.20	41.31	40.22	**42.51**	43.11
		Blue	33.29	30.98	34.74	38.77	36.45	37.26	36.72	38.29	38.02	**39.95**	40.48
02		Red	37.38	36.54	36.83	35.50	38.17	37.29	37.76	36.52	38.19	**38.78**	38.72
		Green	40.94	41.35	41.61	40.57	43.09	42.46	43.81	43.27	44.31	**44.63**	44.64
		Blue	39.60	37.32	40.44	40.05	41.34	40.84	41.35	41.06	42.54	**42.79**	42.81
03		Red	40.21	39.22	40.39	39.19	41.41	40.94	41.58	39.90	41.95	**42.90**	42.91
		Green	42.19	43.15	43.56	41.00	44.75	43.52	45.18	43.78	45.80	**46.06**	46.05
		Blue	39.79	38.37	40.03	38.84	40.83	40.34	41.20	40.30	41.40	**42.29**	42.30
04		Red	36.58	36.73	35.85	35.25	37.01	36.87	37.40	35.92	37.21	**37.77**	37.87
		Green	40.66	42.32	41.47	41.63	43.12	43.81	44.55	44.26	**44.68**	44.53	44.55
		Blue	39.64	38.91	40.74	41.91	41.93	42.33	42.46	42.32	**43.61**	43.42	43.38
05		Red	34.65	32.60	34.95	34.60	36.87	36.87	37.12	36.10	**37.62**	37.03	38.06
		Green	35.81	36.75	37.27	36.78	39.45	39.69	40.02	39.61	**40.88**	40.03	41.05
		Blue	34.27	32.25	34.27	34.26	36.18	36.06	36.42	35.72	**36.71**	36.29	37.17
06		Red	34.66	32.49	37.35	39.02	39.05	38.22	36.86	38.12	40.13	**40.86**	40.93
		Green	35.93	36.91	38.88	42.22	40.81	41.48	39.86	41.68	42.33	**43.51**	43.76
		Blue	34.24	31.98	36.49	38.00	38.00	37.35	36.37	37.43	38.80	**39.34**	39.33
07		Red	40.64	38.77	39.99	39.25	41.26	41.25	41.87	40.07	41.83	**42.64**	42.68
		Green	42.17	42.16	42.64	41.22	43.98	43.96	45.31	43.97	45.27	**45.50**	45.52
		Blue	40.25	38.51	39.37	38.75	40.56	40.69	41.32	39.91	41.01	**41.82**	41.86
08		Red	31.57	28.05	33.06	34.55	34.51	34.56	33.34	34.44	35.08	**36.10**	36.08
		Green	33.37	32.96	35.19	38.45	37.17	38.55	36.93	38.63	38.53	**40.00**	40.07
		Blue	31.55	27.83	33.11	35.28	34.63	34.67	33.46	34.82	35.23	**36.26**	36.32
09		Red	39.38	36.78	40.12	39.72	41.02	40.64	40.99	40.29	41.69	**42.31**	42.30
		Green	41.39	41.40	42.83	42.02	44.18	43.42	43.94	43.90	45.14	**45.44**	45.42
		Blue	40.23	37.39	40.45	40.90	42.78	41.90	40.92	40.91	43.00	**43.30**	43.31
10		Red	39.12	37.64	39.43	40.25	40.79	40.78	40.83	39.94	41.19	**41.63**	41.76
		Green	41.33	42.27	42.80	43.91	44.38	44.38	44.42	44.41	45.37	**45.47**	45.53
		Blue	39.49	37.69	39.89	41.45	41.75	41.41	40.96	40.67	42.08	**42.10**	42.19
11		Red	35.50	33.71	36.44	37.43	38.11	37.96	37.71	37.69	38.75	**39.21**	38.97
		Green	37.03	37.90	38.76	41.20	40.67	41.64	41.25	42.15	42.03	**42.74**	42.81
		Blue	35.87	33.65	37.25	39.03	38.83	38.92	38.70	39.32	39.87	**40.27**	40.16
12		Red	39.89	37.44	40.45	40.89	41.80	41.41	40.97	40.45	42.09	**42.86**	42.87
		Green	42.27	42.41	43.82	44.38	45.23	45.22	44.74	44.82	46.30	**46.63**	46.64
		Blue	40.21	37.74	41.11	42.21	42.43	42.09	41.31	41.46	42.98	**43.38**	43.41
13		Red	29.53	28.86	31.32	36.00	33.15	34.04	34.07	35.84	34.98	**36.48**	36.48
		Green	30.58	32.53	32.18	**38.38**	34.26	36.83	36.02	38.13	36.09	38.14	38.14
		Blue	29.10	28.36	30.43	34.05	32.19	32.86	32.97	34.00	33.56	**34.40**	34.37
14		Red	34.81	33.74	33.82	31.59	35.39	34.57	**35.68**	33.77	35.53	35.54	36.53
		Green	37.14	37.72	37.73	34.98	39.23	38.04	40.25	39.00	**40.28**	39.74	40.55
		Blue	35.02	33.31	34.62	32.64	35.75	35.11	**36.42**	35.11	36.25	36.33	37.03
15		Red	36.08	36.50	35.69	35.68	37.03	36.79	36.95	36.10	37.22	**37.71**	37.74
		Green	39.51	41.56	40.62	40.59	42.05	42.29	42.79	42.32	**43.23**	43.11	43.13
		Blue	37.96	37.57	38.82	39.78	40.02	40.22	40.59	40.62	**41.24**	41.19	41.18
16		Red	38.09	35.44	41.20	42.11	42.69	41.56	39.73	41.14	43.60	**43.81**	44.23
		Green	39.55	40.01	42.66	45.46	44.37	44.82	42.87	44.60	45.75	**46.15**	46.83
		Blue	37.88	35.30	40.41	41.08	41.81	40.85	39.49	40.49	42.49	**42.73**	42.84
17		Red	38.41	36.95	38.93	40.88	40.17	40.79	40.77	40.06	41.38	**41.41**	41.38
		Green	39.19	40.38	40.36	43.17	42.02	43.03	42.94	43.22	43.15	**43.39**	43.41
		Blue	37.74	36.67	38.32	40.52	40.03	40.31	39.92	39.80	40.83	**40.84**	40.81

(*Continued*)

Table I. (Continued)

			HA	LI	HD	SA	DFPD	AP	CCA	CCA+PP	DLMMSE	LPA-ICI	Oracle Γ
18		Red	33.91	33.40	34.21	35.32	35.68	36.23	36.56	35.90	**36.69**	36.56	36.75
		Green	35.24	36.92	36.12	38.36	38.12	39.46	39.42	**39.73**	39.41	39.53	39.54
		Blue	33.75	33.00	34.26	36.41	36.29	36.86	36.71	36.88	**37.27**	37.18	37.23
19		Red	36.67	32.61	37.60	38.67	38.78	38.70	37.90	38.41	39.83	**40.32**	40.32
		Green	38.34	37.24	39.51	42.07	41.28	42.12	41.08	42.25	42.77	**43.36**	43.53
		Blue	37.09	32.64	37.86	40.17	40.01	39.80	38.16	38.85	40.93	**41.48**	41.57
20		Red	38.75	36.98	39.29	40.54	40.50	40.96	41.17	40.66	41.80	**41.90**	41.97
		Green	39.82	40.57	40.85	42.79	42.40	43.50	43.76	43.87	43.86	**44.01**	44.32
		Blue	37.31	35.72	37.55	38.12	38.56	38.71	39.24	38.98	39.27	**39.62**	39.68
21		Red	35.04	33.40	36.46	38.86	37.61	38.47	38.31	39.14	39.14	**39.68**	40.00
		Green	36.29	37.47	37.76	41.92	39.47	41.57	40.90	42.32	41.22	**41.97**	42.45
		Blue	34.48	32.74	35.41	37.50	36.67	37.19	37.16	37.70	37.65	**38.03**	38.09
22		Red	35.80	34.90	35.41	36.53	36.73	37.03	37.22	36.06	37.60	**37.64**	37.73
		Green	37.72	38.44	38.22	38.69	39.55	39.72	40.58	39.88	**40.86**	40.83	40.91
		Blue	35.60	33.84	35.48	36.18	36.75	36.71	37.13	36.55	37.42	**37.55**	37.67
23		Red	41.06	39.83	40.31	38.50	41.26	40.76	41.48	39.25	41.78	**42.76**	42.78
		Green	43.30	43.71	43.92	41.44	44.90	44.03	45.60	44.03	46.24	**46.39**	46.39
		Blue	41.54	40.27	41.15	39.51	41.98	41.61	42.40	41.02	42.78	**43.37**	43.39
24		Red	32.63	32.24	32.70	34.70	34.43	34.94	34.50	33.53	**35.94**	35.62	35.62
		Green	33.59	35.61	35.15	37.38	36.74	37.49	37.45	37.41	38.01	**38.12**	38.14
		Blue	30.76	30.47	31.90	33.03	32.83	32.93	32.95	32.82	**33.74**	33.66	33.67
Mean PSNR		Red	36.40	35.82	36.93	37.58	38.32	38.26	38.21	37.77	39.11	**39.63**	39.77
		Green	38.25	39.06	39.58	40.81	41.22	41.73	41.79	42.02	42.57	**43.00**	43.19
		Blue	36.53	34.69	37.26	38.27	38.69	38.63	38.51	38.54	39.53	**39.90**	40.01

[a] HA (Hamilton and Adams, 1997), LI (Malvar et al., 2004), HD (Hirakawa and Parks, 2005), SA (Li, 2005), DFPD (Menon et al., 2007), AP (Gunturk et al., 2002), CCA (Lukac et al., 2004b), CCA+PP is a demosaicing approach (Lukac et al., 2004b) with postprocessing (Lukac et al., 2004a), DLMMSE-based interpolation (Zhang and Wu, 2005), proposed LPA-ICI-based interpolationin, ''Oracle Γ'' is the proposed LPA-ICI-based interpolation with the optimal threshold parameter Γ.

$\in X_{G_1}$ we have only horizontal $(G_1 + \tilde{R}_h)$ and $(G_1 - \tilde{R}_h)$, and vertical $(G_1 + \hat{B}_v)$ and $(G_1 - \tilde{B}_v)$:

$$\begin{cases} \tilde{\Phi}^h_{g,r} = G_1 + \tilde{R}_h, \ \tilde{\Delta}^h_{g,r} = G_1 - \tilde{R}_h, \\ \tilde{\Phi}^v_{g,b} = G_1 + \hat{B}_v, \ \tilde{\Delta}^v_{g,b} = G_1 - \tilde{B}_v, \end{cases}$$

and similarly

$$\begin{cases} \tilde{\Phi}^v_{g,r} = G_2 + \tilde{R}_v, \ \hat{\Delta}^v_{g,r} = G_2 - \tilde{R}_v, \\ \tilde{\Phi}^h_{g,b} = G_2 + \tilde{B}_h, \ \hat{\Delta}^h_{g,b} = G_2 - \tilde{B}_h, \end{cases}$$

at G_2 positions. The aggregation of the estimates of $\tilde{\Phi}^h_{g,r}$, $\tilde{\Delta}^h_{g,r}$, $\tilde{\Phi}^v_{g,b}$, $\tilde{\Delta}^v_{g,b}$, $\tilde{\Phi}^v_{g,r}$, $\tilde{\Delta}^v_{g,r}$, $\tilde{\Phi}^h_{g,b}$, and $\tilde{\Delta}^h_{g,b}$, obtained by the LPA-ICI, gives the final denoised estimate of green color component as the weighted mean:

$$\begin{cases} \hat{G} = \frac{1}{2}(\hat{\Phi}^h_{g,r} + \hat{\Delta}^h_{g,r})w^h_{g,r} + \frac{1}{2}(\hat{\Phi}^v_{g,b} + \hat{\Delta}^v_{g,b})w^v_{g,b}, \quad (i,j) \in X_{G_1}, \\ \hat{G} = \frac{1}{2}(\hat{\Phi}^v_{g,r} + \hat{\Delta}^v_{g,r})w^v_{g,r} + \frac{1}{2}(\hat{\Phi}^h_{g,b} + \hat{\Delta}^h_{g,b})w^h_{g,b}, \quad (i,j) \in X_{G_2}. \end{cases} \tag{43}$$

This estimate utilizes both R and B spectral components. Here, $\hat{\Phi}^h_{g,r} = \mathcal{LI}\left\{\tilde{\Phi}^h_{g,r}, \sigma_{\tilde{\Phi}^h_{g,r}}\right\}$, $\hat{\Delta}^h_{g,r} = \mathcal{LI}\left\{\tilde{\Delta}^h_{g,r}, \sigma_{\tilde{\Delta}^h_{g,r}}\right\}$, and all other variables $\hat{\Phi}^v_{g,r}$, $\hat{\Delta}^v_{g,r}$, $\hat{\Phi}^v_{g,b}$, $\hat{\Delta}^v_{g,b}$, $\hat{\Phi}^h_{g,b}$, and $\hat{\Delta}^h_{g,b}$ are defined similarly. The weights $w_{g,r}^{h}$, $w_{g,b}^{v}$, $w_{g,r}^{v}$, and $w_{g,b}^{h}$ are computed using the corresponding variances calculated according to (19):

$$\begin{cases} w^h_{g,r} = \sigma^{-2}_{\hat{\Phi}^h_{g,r}} \Big/ \left(\sigma^{-2}_{\hat{\Phi}^h_{g,r}} + \sigma^{-2}_{\hat{\Phi}^v_{g,b}}\right), \\ \qquad w^v_{g,b} = \sigma^{-2}_{\hat{\Phi}^v_{g,b}} \Big/ \left(\sigma^{-2}_{\hat{\Phi}^h_{g,r}} + \hat{\sigma}^{-2}_{\hat{\Phi}^v_{g,b}}\right), \quad (i,j) \in X_{G_1}, \\ w^v_{g,r} = \sigma^{-2}_{\hat{\Phi}^v_{g,r}} \Big/ \left(\sigma^{-2}_{\hat{\Phi}^v_{g,r}} + \sigma^{-2}_{\hat{\Phi}^h_{g,b}}\right), \\ \qquad w^h_{g,b} = \sigma^{-2}_{\hat{\Phi}^h_{g,b}} \Big/ \left(\sigma^{-2}_{\hat{\Phi}^v_{g,r}} + \sigma^{-2}_{\hat{\Phi}^h_{g,b}}\right), \quad (i,j) \in X_{G_2}. \end{cases}$$

All other steps as *interpolation of R/B colors at B/R positions* and *interpolation of R/B colors at G positions* are identical to those given in Sections II-E and II-F.

IV. RESULTS

A. Simulations for Noiseless Data. The proposed LPA-ICI based CFAI was tested on the Kodak set of color test-images and has been compared with the methods that, to the best of the authors knowledge, demonstrate the state-of-the-art performance (example can be seen in Figure 2). The numerical results are summarized in Table I for each of 24 images and ordered in the ascending order of mean PSNR values (the last row), highlighting the best results with bold face. The diagram of mean PSNR values for each color

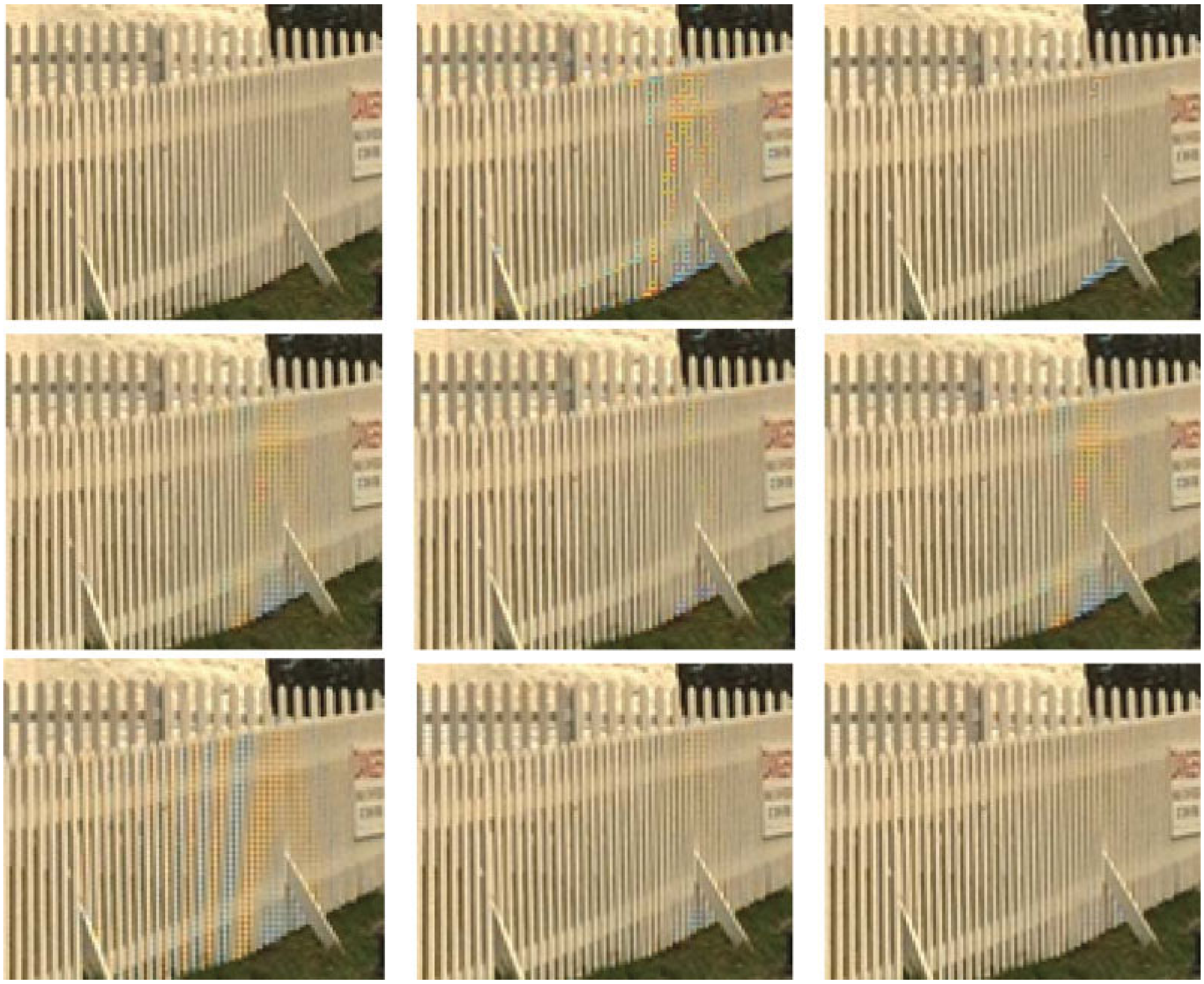


Figure 2. Fragment of the Lighthouse (19) test image (from left to right and from top to bottom): True image; HA (Hamilton and Adams, 1997) PSNR = (36.67 38.34 37.09); HD (Hirakawa and Parks, 2005a) PSNR = (37.60 39.51 37.09); SA (Li, 2005) PSNR = (38.67 42.07 40.17); DFPD (Menon et al., 2007) PSNR = (38.78 42.28 40.01); AP (Gunturk et al., 2002) PSNR = (38.70 42.12 39.80); CCA (Lukac et al., 2004b) PSNR = (37.90 41.08 38.16); DLMMSE-based interpolation (Zhang and Wu, 2005) PSNR = (39.83 42.77 40.93); proposed LPA-ICI-based interpolation PSNR = (40.32 43.36 41.48). [Color figure can be viewed in the online issue, which is available at www.interscience.wiley.com.]

channel is shown in Figure 3. The PSNR values are calculated excluding 15 border pixels to eliminate the boundary effects.

We have used the following parameter settings for the LPA-ICI. Only two directions were used $\Theta = \{0,\pi\}$ for horizontal ''left'' and ''right'' estimates corresponding to the directions of interpolation to obtain $\hat{\Delta}^{h}_{g,r}(i,j)$. For vertical estimates $\hat{\Delta}^{v}_{g,r}(i,j)$, we also used only two vertical ''up'' and vertical ''down'' directions $\Theta = \{\pi/2,3\pi/2\}$. The set S of the scales was $S = \{4,6,8,12\}$.

The threshold Γ in (16) is an important design parameter of the ICI rule. With small Γ the ICI selects only the estimates with the smallest scale s, while with large Γ only the estimates with the largest scale s. The best selection of Γ for each image can be found only if the original images are known. We call these best values of Γ ''Oracles.'' They show that the potential of the developed adaptive algorithm provided the best selection of Γ. The corresponding PSNR values are given in the column ''Oracle Γ'' of Table I. It can be seen that these oracle results are significantly better than the results for all other methods.

We have found an empirical formula giving the image-dependent Γ with the values close to the oracle ones. Let σ_f be standard deviation of high frequency components of G channel calculated as median absolute deviation (MAD) (Donoho, 1995). Then nearly oracle values of the threshold parameter can be calculated as $\Gamma = 0.05\sigma_f + 0.33$. The results with this value of Γ are shown in the ''LPA-ICI'' column of Table I.

It is clearly seen that the proposed technique (''LPA-ICI'' column) gives about 0.4 dB better mean PSNR value (the last row of Table I) than the DLMMSE method (Zhang and Wu, 2005) that demonstrated the best performance among the reviewed CFAI methods. Analyzing the diagram in Figure 3 (left), we can see that this improvement is significant.

The results in terms of average S-CIELAB[†] (Zhang et al., 1997) metric for color images are given in Table II. The techniques are

[2] †The MATLAB code for the S-CIELAB metric is available following the link: http://white.stanford.edu/brian/scielab/scielab.html

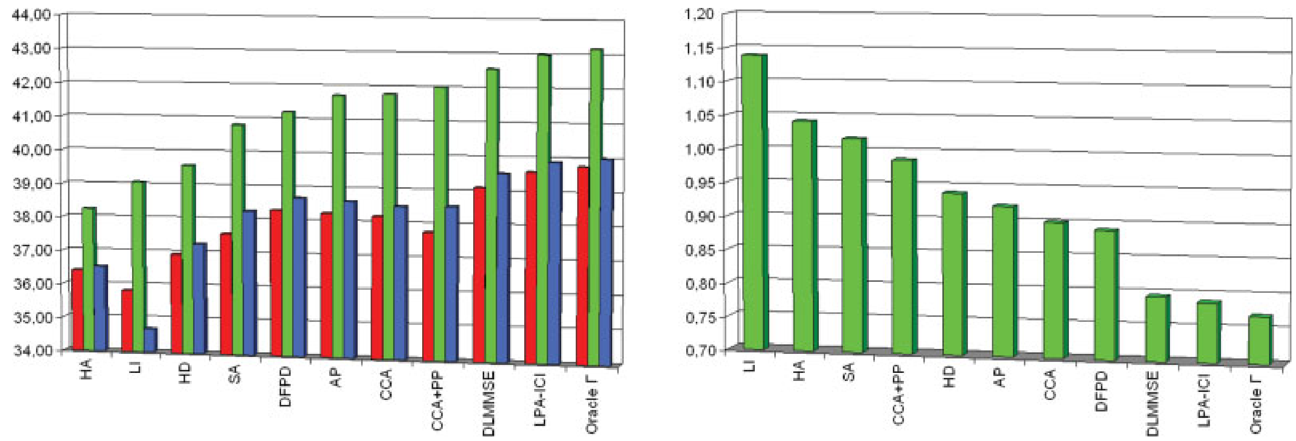


Figure 3. Mean values of PNSR (left) and S-CIELAB (right) for the Kodak test set of 24 images. The following techniques are compared: HA (Hamilton and Adams, 1997), LI (Malvar et al., 2004), HD (Hirakawa and Parks, 2005a), SA (Li, 2005), DFPD (Menon et al., 2007), AP (Gunturk et al., 2002), CCA (Lukac et al., 2004b), CCA+PP is a demosaicing approach (Lukac et al., 2004b) with postprocessing (Lukac et al., 2004a), DLMMSE-based interpolation (Zhang and Wu, 2005), proposed LPA-ICI interpolation, "Oracle Γ" is the proposed LPA-ICI interpolation with the optimal threshold parameter Γ. [Color figure can be viewed in the online issue, which is available at www.interscience.wiley.com.]

presented in the same way as in Table I. The diagram in Figure 3 (right) shows actual ordering of the methods as the S-CIELAB performance is improving. It is seen that the proposed technique provides the best performance for the majority of the test images.

Figure 4 illustrates how the difference $\tilde{\Delta}^{h}_{g,r}$ between the horizontal estimates of green and red color channels is processed for the Lighthouse test image (image number 19 in Table I). It is clearly seen that $\tilde{\Delta}^{h}_{g,r}$ (Fig. 4b) drastically suffers from aliasing comparing it to the true $G - R$ color difference, calculated at G, R lines only (Fig. 4a). Further, the largest errors are near edges and image details (Fig. 4c). We aim to remove these errors by an adaptive filtering using the LPA-ICI in particular. The adaptive scales s obtained with the LPA-ICI filtering for this image (horizontal direction $\theta = 0$ and $\theta = \pi$) are shown in Figures 4e and 4f. It is seen that the filterin

Table II. Average S-CIELAB comparison for different demosaicing methods computed excluding 15 pixels border.[a]

	HA	LI	HD	SA	DFPD	AP	CCA	CCA+PP	DLMMSE	LPA-ICI	Oracle Γ
01	1.6054	2.0037	1.2471	1.2673	1.2927	1.3037	1.2355	1.1731	1.1146	**0.9340**	0.9307
02	0.7306	0.7874	0.8065	0.8989	0.6760	0.7291	0.6998	0.8172	0.6517	**0.6200**	0.6212
03	0.5824	0.5950	0.5746	0.7089	0.5385	0.5764	0.5330	0.6602	0.4817	**0.4562**	0.4559
04	0.8724	0.7934	0.9182	0.9695	0.8198	0.8286	0.8298	0.9768	0.7617	**0.7324**	0.7304
05	1.4005	1.4470	1.3127	1.6829	1.1474	1.2184	1.2283	1.4346	**1.0777**	1.3039	1.0043
06	1.1741	1.4652	0.8661	0.9356	0.8746	0.9968	1.0593	1.0150	0.7807	**0.7645**	0.7871
07	0.6547	0.6391	0.6855	0.8664	0.6254	0.6519	0.5662	0.7788	0.5755	**0.5231**	0.5203
08	1.7936	2.5718	1.4410	1.6281	1.4788	1.5616	1.5606	1.4692	1.3405	**1.2306**	1.2500
09	0.7139	0.7804	0.6338	0.7537	0.5996	0.6727	0.6984	0.8272	0.5624	**0.5453**	0.5461
10	0.6941	0.7007	0.6252	0.6517	0.5942	0.6279	0.6857	0.8063	**0.5518**	0.5568	0.5593
11	1.0218	1.2103	0.8602	0.9100	0.8127	0.8641	0.8740	0.8832	0.7422	**0.7307**	0.7561
12	0.5949	0.6951	0.5238	0.5582	0.5310	0.5522	0.5551	0.6416	0.4645	**0.4472**	0.4484
13	2.4227	2.3780	1.9242	1.6997	1.9206	1.8496	1.6781	1.5134	1.5697	**1.5591**	1.5620
14	1.2053	1.2235	1.2159	1.5656	1.0913	1.1856	1.0195	1.2688	**0.9941**	0.9973	0.9011
15	0.7577	0.6984	0.7917	0.8268	0.7346	0.7467	0.7175	0.8143	0.6615	**0.6413**	0.6338
16	0.8340	1.0404	0.5908	0.6640	0.6011	0.6890	0.7590	0.7580	0.5365	**0.5338**	0.5400
17	0.7263	0.7083	0.6231	0.6006	0.6150	0.6085	0.6187	0.7110	**0.5367**	0.5491	0.5694
18	1.2908	1.1953	1.2608	1.3182	1.1196	1.0856	1.1078	1.2159	**1.0002**	1.0759	1.0559
19	0.9997	1.3028	0.8687	0.8848	0.8274	0.8581	0.9345	0.9930	0.7422	**0.7143**	0.7286
20	0.6910	0.7206	0.6357	0.6565	0.6878	0.6750	0.6011	0.6997	0.5447	**0.5339**	0.5477
21	1.2131	1.3489	1.0246	1.0102	1.0457	1.0170	0.9735	0.9745	0.9051	**0.8892**	0.9428
22	1.1436	1.1679	1.2309	1.2575	1.0807	1.0882	1.0402	1.2314	**0.9747**	0.9959	0.9670
23	0.5639	0.5293	0.6173	0.7604	0.5549	0.5633	0.5452	0.7452	0.4982	**0.4663**	0.4644
24	1.3114	1.3153	1.2729	1.3299	1.2016	1.1794	1.1561	1.2817	**1.0098**	1.1153	1.1212
Mean S-CIELAB	1.0416	1.1382	0.9396	1.0169	0.8946	0.9221	0.9032	0.9871	0.7949	**0.7882**	0.7768

[a] HA (Hamilton and Adams, 1997), LI (Malvar et al., 2004), HD (Hirakawa and Parks, 2005), SA (Li, 2005), DFPD (Menon et al., 2007), AP (Gunturk et al., 2002), CCA (Lukac et al., 2004b), CCA+PP is a demosaicing approach (Lukac et al., 2004b) with postprocessing (Lukac et al., 2004a), DLMMSE-based interpolation (Zhang and Wu, 2005), proposed LPA-ICI-based interpolationin, ''Oracle Γ'' is the proposed LPA-ICI-based interpolation with the optimal threshold parameter Γ.

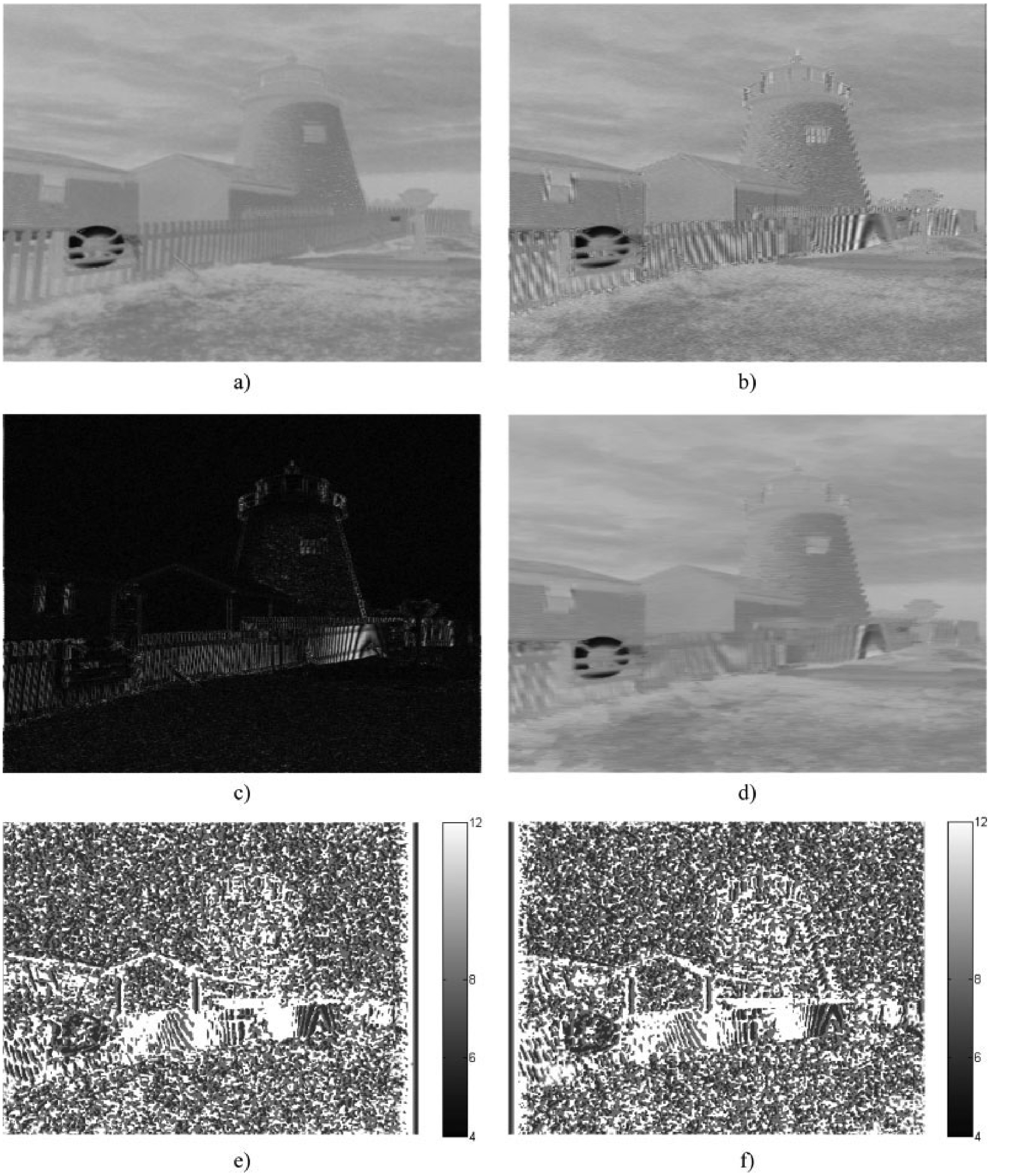


Figure 4. The horizontal difference between *G* and *R* colors: (a) The true difference between the values *R* and *G*; (b) The difference $\tilde{\Delta}^{h}_{g,r}$ between the true values *R* and *G*, and the directional color estimates G_h and R_h; (c) The absolute values of the errors $\varepsilon_{g,r}^{h}$; (d) The filtered difference $\tilde{\Delta}^{v}_{g,r}$ with LPA-ICI in horizontal direction; (e) The LPA-ICI scales ($\theta = 0$) of the filtered $\tilde{\Delta}^{h}_{g,r}$ (right); (f) The LPA-ICI scales ($\theta = \pi$) of the filtered $\tilde{\Delta}^{h}_{g,r}$ (left).

performed horizontally selects smaller scales near vertical details that allows to avoid oversmoothing of these details. The filtered difference $\hat{\Delta}^{h}_{g,r}$ is shown in Figure 4d.

Our study shows that the ''demosaicing noise'' is not white and strongly localized. At different parts of an image, the power of noise is different. It justifies the use of the local estimates of the

Table III. PSNR values for CFA interpolation of images corrupted by Gaussian noise. [Color figure can be viewed in the online issue, which is available at www.interscience.wiley.com.]

		07	08	13	19	23	Mean PSNR
HA (Hamilton and Adams, 1997) with prefiltering (Katkovnik et al., 2002)	R	30.29	26.50	24.56	29.11	31.93	28.48
	G	31.06	27.30	24.79	29.94	32.66	29.15
	B	29.60	26.17	24.69	29.34	30.77	28.11
AP (Gunturk et al., 2002) with prefiltering (Katkovnik et al., 2002)	R	30.51	27.24	25.25	29.35	31.96	28.86
	G	31.06	27.97	25.37	30.11	32.60	29.42
	B	30.53	27.39	25.65	30.09	31.93	29.12
DLMMSE (Zhang and Wu, 2005) with prefiltering (Katkovnik et al., 2002)	R	30.89	27.51	25.40	29.66	32.34	29.16
	G	31.48	28.19	25.39	30.36	33.08	29.70
	B	31.00	27.65	25.75	30.36	32.39	29.43
Proposed LPA-ICI based integrated demosaicing with denoising	R	**32.10**	**27.81**	**27.64**	**30.11**	**33.20**	**30.18**
	G	**32.47**	**28.70**	**27.73**	**30.86**	**34.15**	**30.79**
	B	**32.05**	**27.91**	**27.53**	**30.76**	**33.36**	**30.32**

variance in (40). As a result, suppression of color distortions becomes much better in terms of both numerical and visual evaluation.

Visual comparison of different methods is presented in Figure 2 for the fragment of the Lighthouse test image (19). The color artifacts are removed almost completely by the proposed method (Fig. 2 bottom right image). It is done significantly better than by other methods.

The computational complexity of the proposed technique is relatively high. For instance, it is higher than the complexity of DLMMSE, which provides the second best results among the considered techniques, since the filter supports are larger and multiple scales are used for each direction. The adaptivity to data requires additional calculations. However, the complexity evaluation in terms of processing time calculated for several benchmark techniques shows that the proposed technique provides competitive results and efficient implementation is possible. In particular, the average time for processing a 512 × 768 image by HA CFAI was 1.2 and 3.3 s for CCA, 4.2 s for CCA+PP[‡], 53 s for DLMMSE, and 12 s for the proposed LPA-ICI-based CFAI. The simulations were performed, and proposed algorithm was implemented in MATLAB environment (version 7.1 SP3). The PC used for simulations was Pentium 4 HT 3.2 GHz (CPU), 2 GB of RAM, and OS Windows XP SP2.

B. Simulations for Noisy Data. We have used the standard test images with the intensities in the range of 0–255 in our simulations for the proposed integrated CFAI and denoising technique. The following three noise models are considered:

a. The additive stationary white Gaussian noise with the invariant standard deviation $\sigma(i,j)$ constant for all $(i,j) \in X$ and for all color intensities G_1, G_2, R, and B;
b. The signal-dependent Poissonian observations with $\chi z_{\text{bayer}}(i,j) \sim \mathscr{P}(\chi\Psi\{y_{RGB}(i,j)\})$. This noise can be written explicitly in the additive form (1) where the standard deviation depends on the image intensity as $\sigma_{\text{bayer}}(i,j) = \text{std}\{z_{\text{bayer}}(i,j)\} = \sqrt{\Psi\{y_{RGB}(i,j)\}/\chi}$. It is shown in (Foi et al., 2006, 2007) that such a model can be used for generic CMOS digital imaging sensors.
c. The nonstationary Gaussian noise with the signal-dependant standard deviation $\sigma_{\text{bayer}}(i,j) = k_0 + k_1\ \Psi\{y_{RGB}(i,j)\}$ (Hirakawa and Parks, 2005b, 2006).

For the presented experiments, we have used $\sigma(i,j) = 12.75$ for the Gaussian model (a), $\chi = 0.5447$ for the Poissonian model (b), and considered two cases for (c): $k_0 = 25$, $k_1 = 0$, and $k_0 = 10$, $k_1 = 0.1$. The simulations for (c) are targeted to make an objective comparison with the method proposed in (Hirakawa and Parks, 2005b, 2006) for noisy data (27).

The LPA-ICI filtering for noisy data was exploited for the eight directions $\theta \in \Theta = \{k \cdot 2\pi/8\colon k = 0, \ldots, 7\}$ with the scale (window size) values given by the set $S = \{1, 2, 4, 7, 10\}$, which is essentially different from settings for noiseless data. The threshold parameter Γ of the ICI rule was fixed as $\Gamma = 1$ for filtering the sums and as $\Gamma = 1.5$ for filtering the differences.

The results are shown in Tables III and IV for five test images of the sizes 512 × 768. The PSNR values were calculated excluding 15 border pixels to eliminate boundary effects. The mean PSNR values calculated over the five test images are given in the last column of the table (Mean PSNR).

The PSNR values for the case of the Gaussian noise (a) are shown in Table III. The proposed algorithm is compared versus: Hamilton–Adams (''HA'') (Hamilton and Adams, 1997), Alternating Projections (''AP'') (Gunturk et al., 2002), and the DLMMSE algorithm (''DLMMSE'') (Zhang and Wu, 2005) with the LPA-ICI prefiltering of the noisy data (Katkovnik et al., 2002). This prefiltering was applied to each color channel independently as the preprocessing stage. It is seen that in average (the last column of the table) the proposed integrated denoising and demosaicing technique provides PSNR at least 1 dB better than other algorithms. The LPA-ICI prefiltering for the Hamilton–Adams, DLMMSE, and Alternating Projections algorithm have been chosen to make an objective comparison between the proposed algorithm designed for noisy data and the considered alternatives known as very good demosaicing algorithms designed for noiseless data.

[‡] The Windows executable files for the CCA and CCA+PP techniques were downloaded from http://www.dsp.utoronto.ca/~lukacr/. The implementation environment is unknown.

Table IV. PSNR values for CFA interpolation of images corrupted by Poissonian noise. [Color figure can be viewed in the online issue, which is available at www.interscience.wiley.com.]

		07	08	13	19	23	Mean PSNR
HA (Hamilton and Adams, 1997) with prefiltering (Foi et al., 2005)	R	30.07	25.72	25.00	28.17	31.59	28.11
	G	31.01	26.61	25.60	29.07	32.57	28.96
	B	29.87	25.42	24.93	28.61	30.79	27.92
AP (Gunturk et al., 2002) with prefiltering (Foi et al., 2005)	R	30.42	26.49	26.07	28.53	31.85	28.67
	G	31.13	27.16	26.55	29.33	32.56	29.34
	B	30.77	26.56	26.26	29.30	31.85	28.94
DLMMSE (Zhang and Wu, 2005) with prefiltering (Foi et al., 2005)	R	30.75	26.67	26.34	28.68	32.11	28.91
	G	31.53	27.45	26.66	29.52	33.02	29.63
	B	31.29	26.81	26.52	29.59	32.49	29.33
Proposed LPA-ICI based integrated demosaicing with denoising	R	**31.76**	**27.17**	**27.18**	**29.39**	**32.59**	**29.62**
	G	**32.36**	**27.94**	**27.29**	**30.13**	**33.42**	**30.23**
	B	**32.08**	**27.28**	**27.10**	**30.15**	**32.85**	**29.89**

The PSNR values for restoration of Bayer data corrupted by signal-dependant Poissonian type of noise (b) are given in Table IV. For comparison, we used the same alternative algorithms as those used for the Gaussian noise. A special recursive version of the LPA-ICI filtering developed specifically for observations with signal-dependent noise (Foi et al., 2005; Katkovnik et al., 2006) was applied as prefiltering. The results are summarized in Table IV, where four iterations of this recursive prefiltering have been carried out. It is seen from Table IV that the proposed denoising–interpolation technique provides at least 0.6 dB better PSNR values than the alternative algorithms with the noise prefiltering. Note, that the proposed algorithm is nonrecursive.

Visual comparison of the results is important for algorithm evaluation. Figures 5 and 6 illustrate some difficult parts of the restored

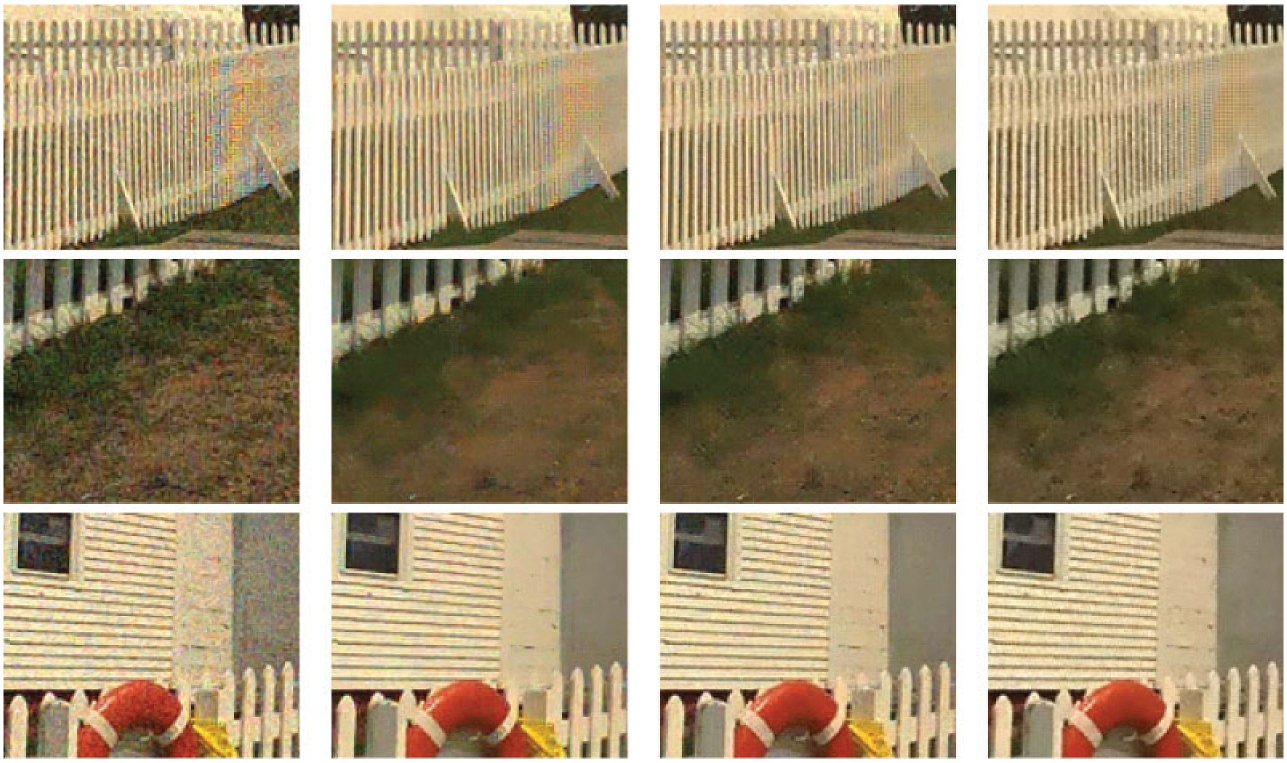


Figure 5. Restoration of the Lighthouse test image corrupted by Gaussian noise with σ = 0.05. Columns are enumerated from left to right: interpolated noisy image by HA (Hamilton and Adams, 1997) CFAI; restoration by HA (Hamilton and Adams, 1997) CFAI with LPA-ICI (Katkovnik et al., 2002) denoising at the prefiltering step, PSNR = (29.11, 29.94, 29.34); restoration by DLMMSE (Zhang and Wu, 2005) CFAI with LPA-ICI (Katkovnik et al., 2002) denoising at the prefiltering step, PSNR = (29.66, 30.36, 30.36); proposed LPA-ICI based integrated interpolation and denoising, PSNR = (30.11, 30.86, 30.76). [Color figure can be viewed in the online issue, which is available at www.interscience.wiley.com.]

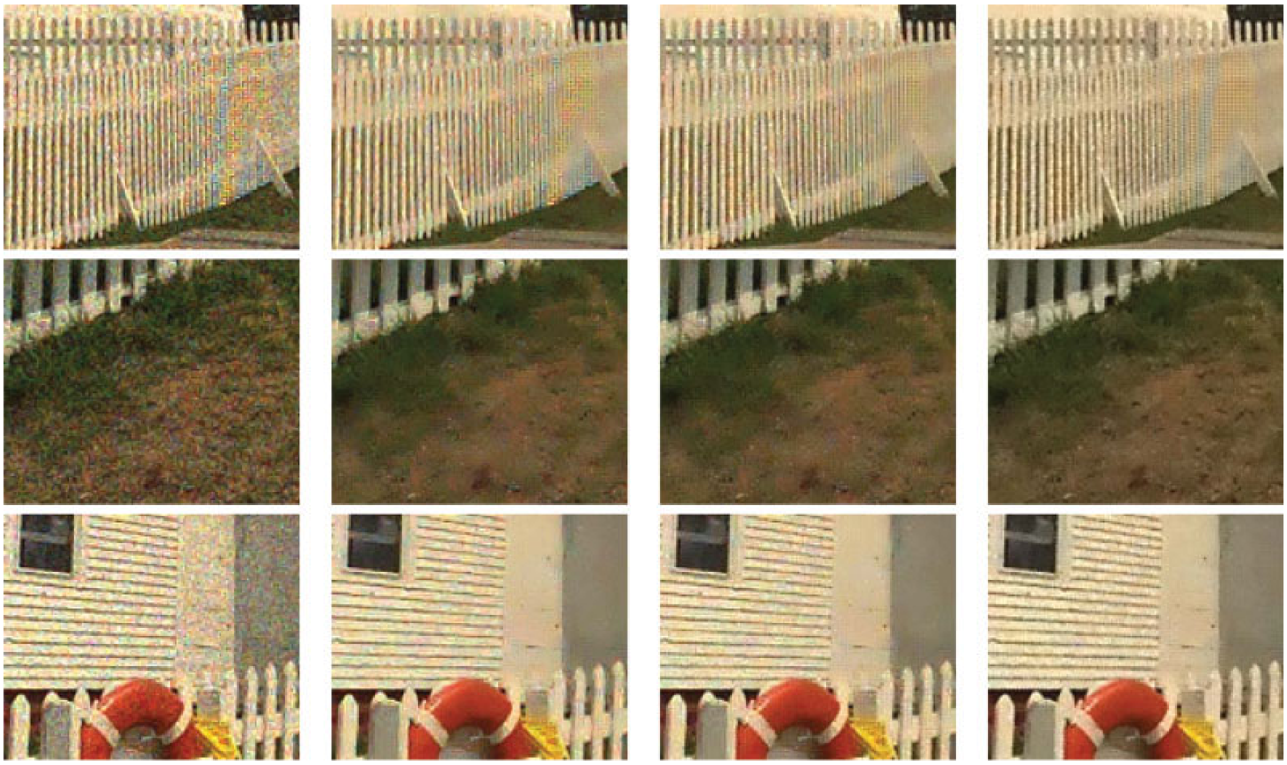


Figure 6. Restoration of the Lighthouse test image corrupted by signal-dependant noise. Columns are enumerated from left to right: interpolated noisy image by HA (Hamilton and Adams, 1997) CFAI; restoration by HA (Hamilton and Adams, 1997) CFAI with iterative LPA-ICI denoising (Foi et al., 2005) at the prefiltering step, PSNR = (28.17, 29.07, 28.61); restoration by DLMMSE (Zhang and Wu, 2005) CFAI with iterative LPA-ICI denoising (Foi et al., 2005) at the prefiltering step, PSNR = (28.68, 29.52, 29.59); proposed LPA-ICI based integrated interpolation with denoising, PSNR = (29.39, 30.13, 30.15). [Color figure can be viewed in the online issue, which is available at www.interscience.wiley.com.]

Lighthouse test-image. Because of the denoising performed independently for each color channels, the final image visually looks oversmoothed and suffers from color artifacts visible especially near edges (second and third columns), even for very advanced CFAI techniques. In combination with aliasing problem (noticeable at the fence and wall regions of the Lighthouse image), the color artifacts become visible significantly. In the case of Poissonian noise, this problem becomes even stronger. It is seen that the proposed technique (right column) provides significantly better performance also at the regions that contain small details and textures difficult for restoration.

The LPA-ICI denoising embedded into the interpolation procedure helps to avoid or reduce the above-mentioned problems. As a result, numerical (''LPA-ICI'' row of Tables III and IV) and visual quality evaluation (Figs. 5 and 6, right column) show better performance. The high frequency regions difficult for denoising like the grass region are preserved significantly better and color artifacts are reduced. As a result, the restored image looks more natural.

Table V. PSNR values for CFA interpolation of images corrupted by noise with $\sigma = k_0 + k_1 \Psi \{y_{RGB}\}$ (Hirakawa and Parks, 2005b, 2006).

		07	08	13	19	23	Mean PSNR
Stationary Gaussian Noise				$(k_0, k_1) = (25, 0)$			
Joint demosaicing and denoising (Hirakawa and Parks, 2005b, 2006)	R	27.24	22.44	22.49	24.99	29.70	25.37
	G	27.38	23.59	22.76	26.19	30.94	26.17
	B	26.75	22.72	22.73	26.12	29.51	25.57
Proposed	R	**28.54**	**24.27**	**24.01**	**26.62**	**30.22**	**26.73**
	G	**28.47**	**24.67**	**23.89**	**27.31**	**30.91**	**27.05**
	B	**28.20**	**24.64**	**24.28**	**27.66**	**30.04**	**26.96**
Signal-Dependant Noise				$(k_0, k_1) = (10, 0.1)$			
Joint demosaicing and denoising (Hirakawa and Parks, 2005b, 2006)	R	28.02	22.68	22.98	25.20	29.82	25.74
	G	28.39	23.91	23.31	26.48	31.34	26.69
	B	28.08	23.00	23.26	26.53	30.27	26.23
Proposed	R	**29.43**	**24.64**	**24.48**	**26.88**	**30.49**	**27.18**
	G	**29.79**	**24.96**	**24.38**	**27.61**	**31.37**	**27.62**
	B	**29.61**	**24.90**	**24.64**	**28.05**	**30.56**	**27.55**

Table VI. Average S-CIELAB color metric values for CFA interpolation of images corrupted by noise with $\sigma = k_0 + k_1 \Psi \{y_{RGB}\}$ (Hirakawa and Parks, 2005b, 2006).

	07	08	13	19	23	Mean S-CIELAB
Stationary Gaussian Noise			$(k_0, k_1) = (25, 0)$			
Joint demosaicing and denoising (Hirakawa and Parks, 2005b, 2006)	5.0842	6.9544	7.0168	4.9872	3.7392	5.5564
Proposed	**4.2621**	**5.5945**	**5.9128**	**4.1552**	**3.4694**	**4.6788**
Signal-Dependant Noise			$(k_0, k_1) = (10, 0.1)$			
Joint demosaicing and denoising (Hirakawa and Parks, 2005b, 2006)	4.4298	6.5470	6.3406	4.6225	3.4103	5.0701
Proposed	**3.7884**	**5.4938**	**5.5434**	**3.9609**	**3.2327**	**4.4038**

Comparison with the joint demosaicing and denoising targeted interpolation of data corrupted by signal-dependant noise (c) (Hirakawa and Parks, 2005b, 2006) was performed on five images with the noise parameters as in (Hirakawa and Parks, 2005b). The numerical evaluation is presented in Table V for PSNR values and in Table VI for average S-CIELAB values. It is clearly seen that the proposed technique always significantly outperforms the technique from (Hirakawa and Parks, 2005b, 2006). The visual comparison on the Window (7) and Parrots (23) test images is given in Figure 7 for $k_0 = 10$, $k_1 = 0.1$ for the noise model (c). The HA CFAI (Hamilton and Adams, 1997) was used to visualize the noisiness of the simulated noisy Bayer data (Fig. 7, first column). The second column contains restored fragments by (Hirakawa and Parks, 2005b, 2006) and the third column corresponds to the results obtained by the proposed technique. The proposed technique provides less color artifacts that is supported by the better S-CIELAB values in Table VI. The difference is significant especially for the Window test image.

The restoration of real noisy Bayer data measured directly from the sensor of a cameraphone is illustrated in Figure 8. The noise model and its parameters were identified exactly in the same way how it is done in (Foi et al., 2006, 2007). The left image was interpolated by HA CFAI (Hamilton and Adams, 1997) and the right by the proposed CFAI for noisy data. The histograms for both of them were equalized to improve visual perception in print. No other color correction steps, pre- and postfiltering were done in these experiments.

V. CONCLUSIONS

In this article, we proposed a novel demosaicing technique based on spatially adaptive filtering of interpolation errors. These errors are treated as noise that is nonstationary and strongly depends on signal. The magnitude of errors is higher near edges of image details. We efficiently exploited the adaptive LPA-ICI denoising to remove the errors and preserve the edges. This idea was exploited for

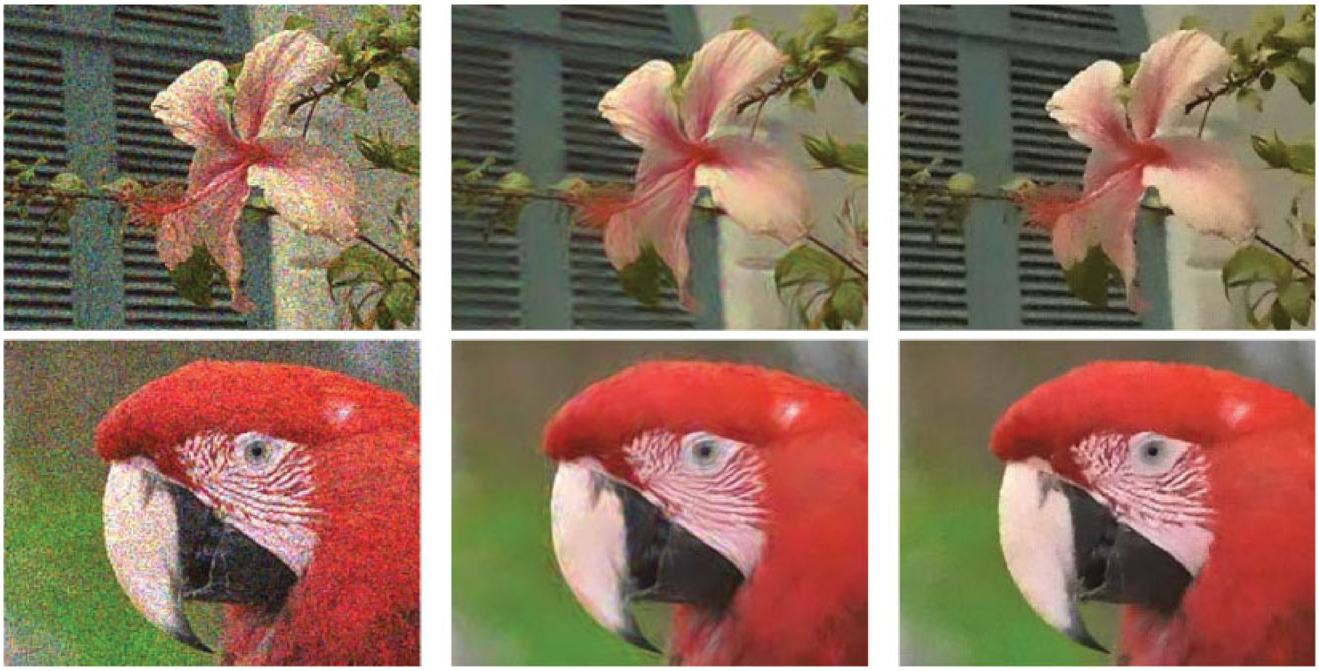


Figure 7. Fragment of the restored Window (7) and Parrots (23) test images corrupted by noise with $\sigma = k_0 + k_1 \Psi\{y_{RGB}\}$ (Hirakawa and Parks, 2005b, 2006), where $k_0 = 10$, $k_1 = 0.1$. Columns are enumerated from left to right: interpolated by HA (Hamilton and Adams, 1997) CFAI noisy image; restoration by joint denoising and demosaicing technique (Hirakawa and Parks, 2005b, 2006); proposed technique. [Color figure can be viewed in the online issue, which is available at www.interscience.wiley.com.]

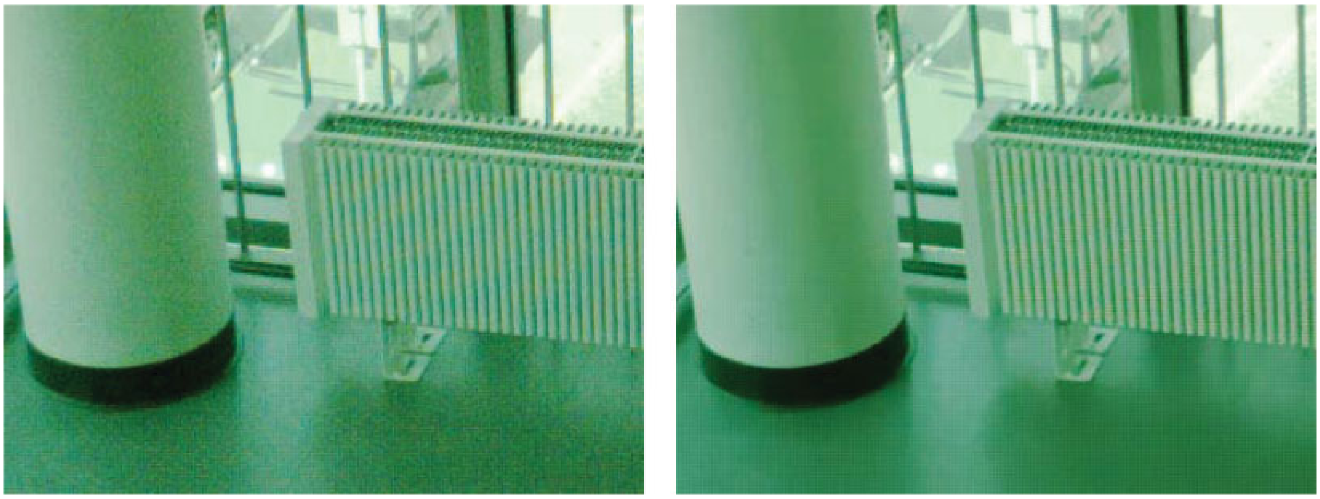


Figure 8. Restoration of raw sensor data taken by cameraphone: HA (Hamilton and Adams, 1997) interpolation (left); the proposed integrated denoising and interpolation (right) for noisy data. [Color figure can be viewed in the online issue, which is available at www.interscience.wiley.com.]

noiseless and noisy Bayer data considering signal-dependant noise natural for CCD/CMOS digital imaging sensors. The CFAI and denoising were integrated into a single procedure that showed significant improvement comparing it with the independent analogous denoising and CFAI. The efficiency and applicability of the proposed technique were shown by simulations for both artificial and real raw data (taken directly from the sensor of a cameraphone).

ACKNOWLEDGMENTS

The authors thank Dr. L. Zhang for providing the implementation code of the technique (Zhang and Wu, 2005) and D. Menon for providing the implementation code of the technique (Menon et al., 2007). We highly appreciate helpful comments of the anonimous reviwers. The authors also thank Dr. A. Foi for useful and practical discussions.

APPENDIX: DESIGN OF DIRECTIONAL LINEAR FILTERS AND INTERPOLATORS IN POLYNOMIAL BASIS

Let $\tilde{X} \subseteq X$ be a domain of coordinates $\tilde{x} \in \tilde{X}$, where the observations $y(\tilde{x})$, $\tilde{x} \in \tilde{X}$ are given. Particularly, if the domain of processing is not subsampled, then $\tilde{X} = X$ that is typical for denoising problems. For green channel $\tilde{X} = X_{G_1} \cup X_{G_2}$. For red and blue, we have $\tilde{X} = X_R$ and $\tilde{X} = X_B$, respectively. It can be used to design interpolation kernels. The interpolation should be produced for the missed pixels $\bar{X} = X/\tilde{X}$ if the data is subsampled $\bar{X} \neq \emptyset$.

It is assumed that $y(x)$ is a piecewise smooth function, which locally can be well-approximated by polynomials (monomials)

$$\frac{1}{i!j!}x_1^i x_2^j, \quad i = 0, ..., m_1, \quad j = 0, ..., m_2.$$

Here, $m = (m_1, m_2)$ is the order of this set of polynomials. The maximal number of the linear independent polynomials of the order m is equal to $M = (m_1 + 1)(m_2 + 1)$.

Let $\phi(x)$ be a set of these linear independent polynomials $\phi_k(x)$ presented as a vector-function

$$\boldsymbol{\phi}(x) = (\phi_0(x), \phi_1(x), ..., \phi_M(x))^T,$$

where "T" denotes the transpose operation.

The polynomials in this vector are ordered according to their power defined for $x_1^i x_2^j$ as $i + j$. For instance

$$\begin{aligned}
&\phi_0 = 1, \quad \text{for } i + j = 0,\\
&\phi_1 = x_1, \qquad \phi_2 = x_2, \qquad \text{for } i + j = 1,\\
&\phi_3 = \frac{x_1^2}{2}, \qquad \phi_4 = \frac{x_2^2}{2}, \qquad \phi_5 = x_1 x_2, \qquad \text{for } i + j = 2,\\
&\phi_6 = \frac{x_1^3}{6}, \quad \phi_7 = \frac{x_2^3}{6}, \quad \phi_8 = \frac{x_1^2 x_2}{2}, \quad \phi_9 = \frac{x_1 x_2^2}{2}, \quad \text{for } i + j = 3.
\end{aligned}$$

The LPA of $y(x)$ at the point $x \in X$ is of the form

$$\hat{y}(x, \tilde{x}) = \mathbf{C}^{\mathrm{T}} \boldsymbol{\phi}(x - \tilde{x}), \tag{A1}$$

where $\mathbf{C} = (C_0, C_1, \ldots, C_M)^{\mathrm{T}}$ are the coefficients of this expansion to be found.

To find the vector $\mathbf{C}$ in (A1), we use the weighted residual quadratic criterion:

$$J_s(x) = \sum_{\tilde{x} \in \tilde{X}} w_s(x - \tilde{x})(z(\tilde{x}) - \mathbf{C}^{\mathrm{T}} \boldsymbol{\phi}(x - \tilde{x}))^2, \quad x \in X, \tag{A2}$$

where $w_s(x)$ is a window function with a scaling parameter s defining the neighborhood size and the residual weights in the LPA.

In particular, we use the nonsymmetric uniform window of the length s_1 and the width s_2 to design denoising kernels (14):

$$w_s(x) = \begin{cases} \frac{1}{s_1(s_2-1)}, & \text{for } 0 \le x_1 < s_1, \ |x_2| < \frac{s_2}{2}, \\ 0, & \text{otherwise}, \end{cases} \tag{A3}$$

where $x \in \tilde{X}$, s_2 is even. We use symmetric uniform window

$$w_s(x) = \begin{cases} \frac{1}{(s_1-1)(s_2-1)}, & \text{for } |x_1| < \frac{s_1}{2}, \ |x_2| < \frac{s_2}{2}, \\ 0, & \text{otherwise}, \end{cases} \tag{A4}$$

to design the interpolation kernels (21),(24) for $x \in \overline{X}$, where $s_1 = s_2$.

It is emphasized that the sets $\overline{X}$ and $\tilde{X}$ are different. The set $\overline{X}$ is a collection of the ''missed'' points where there is no observations and the signal should be interpolated for these points. Contrary to it the set $\tilde{X}$ is a set of the observed points, where values of the signal true or noisy are given.

The estimates of $\mathbf{C}$ are found by minimization of (A2)

$$\hat{\mathbf{C}}(x,s) = \arg\min_{\mathbf{C}} J_s(x).$$

The minimum condition

$$\frac{\partial J_s(x)}{\partial \mathbf{C}^T} = -2\sum_{\tilde{x}\in\tilde{X}} w_s(x-\tilde{x})(z(\tilde{x}) - \mathbf{C}^T\phi(x-\tilde{x}))\phi^T(x-\tilde{x}) = 0$$

gives a system of the linear equations

$$\sum_{\tilde{x}\in\tilde{X}} w_s(x-\tilde{x})z(\tilde{x})\boldsymbol{\phi}^T(x-\tilde{x}) = \mathbf{C}^T\sum_{\tilde{x}\in\tilde{X}} w_s(x-\tilde{x})\boldsymbol{\phi}(x-\tilde{x})\boldsymbol{\phi}^T(x-\tilde{x}). \tag{A5}$$

with the solution

$$\hat{\mathbf{C}}(x,h) = \sum_{\tilde{x}\in\tilde{X}} w_s(x-\tilde{x})\Phi_s^{-1}\boldsymbol{\phi}(x-\tilde{x})z(\tilde{x}), \tag{A6}$$

$$\Phi_h = \sum_{\tilde{x}\in\tilde{X}} w_s(x-\tilde{x})\phi(x-\tilde{x})\boldsymbol{\phi}^T(x-\tilde{x}). \tag{A7}$$

Substituting $\hat{\mathbf{C}}(x,s)$ into (A1), we obtain the polynomial estimate of the signal $\hat{y}_s(x,\tilde{x}) = \boldsymbol{\phi}^T\,(x-\tilde{x})\,\hat{\mathbf{C}}(x,s)$ valid in a neighborhood of the point $x \in X$. According to the idea of the LPA, we use this model only for the center of the LPA, i.e., for $x = \tilde{x}$.

Then the estimate $\hat{y}_s(x,\tilde{x})$ is transformed to the final form

$$\hat{y}_s(x) = \hat{y}_s(x,x) = \hat{\mathbf{C}}^{\mathrm{T}}(x,s)\boldsymbol{\phi}(0) = \\ \sum_{\tilde{x}\in\tilde{X}} w_s(x-\tilde{x})\boldsymbol{\phi}^T(x-\tilde{x})\Phi_s^{-1}\boldsymbol{\phi}(0)z(\tilde{x}).$$

This interpolation estimate can be rewritten in the form of convolution

$$\hat{y}_s(x) = \sum_{\tilde{x}\in\tilde{X}} g_s(x-\tilde{x})z(\tilde{x}), \quad x \in X, \qquad \tilde{x} \in \tilde{X}, \tag{A8}$$

with the convolution kernel

$$g_s(x) = w_s(x)\boldsymbol{\phi}^{\mathrm{T}}(x)\Phi_s^{-1}\boldsymbol{\phi}(0), \tag{A9}$$

because the window $w_s(x) = 0$ for $x = \bar{x}$. Thus, the kernel $g_s(x)$ is also equal to 0 for $x = \bar{x}$, $g_s(\bar{x}) = 0$.

Note that the kernels g_s essentially depend on a given interpolation grid.

Using the directional windows $w_{s,\theta}$ in (A3) and (A4), we obtain directional kernels $g_{s,\theta}$ (Katkovnik et al., 2006).

In the case of interpolation, the kernels (A9) are essentially different from the standard LPA kernels (Katkovnik et al., 2006) by zeros used to fill the kernel support at the positions of the missed observations.

REFERENCES

J.E. Adams Jr., Design of color filter array interpolation algorithms for digital cameras, Part 2, IEEE Proc Int Conf Image Process 1 (1998), 488–492.

J. Adams, K. Parulski, and K. Spaulding, Color processing in digital cameras, IEEE Micro 18 (1998), 20–30.

D. Alleysson, S. Susstrunk, and J. Herault, Color demosaicing by estimating luminance and opponent chromatic signals in the Fourier domain, Proc IS&T/SID 10th Color Imaging Conf, 2002, pp. 331–336.

D. Alleysson, S. Susstrunk, and J. Marguier, Linear demosaicing inspired by the human visual system, IEEE Trans Image Process 14 (2005), 439–449.

B.E. Bayer, Color imaging array, U.S. Patent 3,971,065, 1976.

D.H. Brainard, Bayesian method for reconstructing color images from trichromatic samples, IS&T's 47th Annual Conference/ICSP, 1994, 375–380.

D.R. Cok, Reconstruction of CCD images using template matching, IS&T's 47th Annual Conference/ICPS, 1994, pp. 380–385.

D.L. Donoho, De-noising by soft-thresholding, IEEE Trans Inform Theory 41 (1995), 613–627.

E. Dubois, Frequency-domain methods for demosaicking of Bayer-sampled color images, IEEE Signal Process Lett 12 (2005), 847–850.

A. Foi, S. Alenius, V. Katkovnik, and K. Egiazarian, Noise measurement for raw-data of digital imaging sensors by automatic segmentation of non-uniform targets, IEEE Sensors J 2007 (7), 1456–1461.

A. Foi, R. Bilcu, V. Katkovnik, and K. Egiazarian, Anisotropic local approximations for pointwise adaptive signal-dependent noise removal, Proc XIII European Signal Process Conf, EUSIPCO 2005, Antalya, Turkey, September 4–8, 2005.

A. Foi, V. Katkovnik, D. Paliy, K. Egiazarian, M. Trimeche, S. Alenius, R. Bilcu, and M. Vehvilainen, Apparatus, method, mobile station and computer program product for noise estimation, modeling and filtering of a digital image, U.S. Patent Application No. 11/426,128, 2006.

B.K. Gunturk, Y. Altunbasak, and R.M. Mersereau, Color plane interpolation using alternating projections, IEEE Trans Image Process 11 (2002), 997–1013.

B.K. Gunturk, J. Glotzbach, Y. Altunbasak, R. Schafer, and R.M. Mersereau, Demosaicking: Color filter array interpolation, IEEE Signal Process Mag 22 (2005), 44–54.

J.F. Hamilton Jr. and J.E. Adams, Adaptive color plane interpolation in single color electronic camera, U.S. Patent 5,629,734, May 1997.

K. Hirakawa and T.W. Parks, Adaptive homogeneity-directed demosaicing algorithm, IEEE Trans Image Process 14 (2005a), 360–369.

K. Hirakawa and T.W. Parks, Joint demosacing and denoising, IEEE ICIP III (2005b), 309–312.

K. Hirakawa and T.W. Parks, Joint demosaicing and denoising, IEEE Trans Image Process 15 (2006), 2146–2157.

O. Kalevo and H. Rantanen, Noise reduction techniques for Bayer-matrix images, sensors and camera systems for scientific, industrial, and digital photography applications III, Proceedings of SPIE 4669, 2002.

V. Katkovnik, A new method for varying adaptive bandwidth selection, IEEE Trans Signal Process 47 (1999), 2567–2571.

V. Katkovnik, K. Egiazarian, and J. Astola, Adaptive window size image de-noising based on intersection of confidence intervals (ICI) rule, J Math Imaging Vis 16 (2002), 223–235.

V. Katkovnik, K. Egiazarian, and J. Astola, Local approximation techniques in signal and image processing, Monograph Vol. PM157, SPIE Press, 2006.

D. Keren and M. Osadchy, Restoring subsampled color images, Machine Vis Appl 11 (1999), 197–202.

R. Kimmel, Demosaicing: Image reconstruction from color CCD samples, IEEE Trans Image Process 8 (1999), 1221–1228.

C.A. Laroche and M.A. Prescott, Apparatus and method for adaptively interpolating a full color image utilizing chrominance gradients, U.S. Patent 5,373,322, 1994.

X. Li, Demosaicing by successive approximation, IEEE Trans Image Process 14 (2005), 370–379.

R. Lukac, K. Martin, and K.N. Plataniotis, Demosaicked image postprocessing using local color ratios, IEEE Trans Circuits Sys Video Technol 14 (2004a), 914–920.

R. Lukac and K.N. Plataniotis, An efficient CFA interpolation solution, 46th Int Symposium Electronics in Marine, ELMAR, 2004a.

R. Lukac and K.N. Plataniotis, Normalized color-ratio modeling for CFA interpolation, IEEE Trans Consum Electron 50 (2004b), 737–745.

R. Lukac and K.N. Plataniotis, Color filter arrays: Design and performance analysis, IEEE Trans Consum Electron 51 (2005a), 1260–1267.

R. Lukac and K.N. Plataniotis, Data-adaptive filters for demosaicking: A framework, IEEE Trans Consum Electron 51 (2005b), 560–570.

R. Lukac and K.N. Plataniotis, Demosaicking using vector spectral model, Proceedings of the 2006 IEEE International Conference on Multimedia and Expo ICME, 2006a, pp. 1185–1188.

R. Lukac and K.N. Plataniotis, ‘‘Single-sensor camera image processing,’’ In Color image processing: Methods and applications, R. Lukac and K.N. Plataniotis (editors), CRC Press/Taylor & Francis, 2006b, 363–392.

R. Lukac, K.N. Plataniotis, D. Hatzinakos, and M. Aleksic, A novel cost effective demosaicing approach, IEEE Trans Consum Electron 50 (2004b), 256–261.

R. Lukac, B. Smolka, K. Martin, K.N. Plataniotis, and A.N. Venetsanopulos, Vector filtering for color imaging, IEEE Signal Process Mag 22 (2005), 74–86.

H. S. Malvar, L.-W. He, and R. Cutler, High-quality linear interpolation for demosaicing of Bayer-patterned color images, IEEE Int Conf (ICASSP ’04), Proc Acoust Speech Signal Process 3 (2004), 485–488.

D. Menon, S. Andriani, and G. Calvagno, Demosaicing with directional filtering and a posteriory decision, IEEE Trans Image Process 16 (2007), 132–141.

D. Paliy, R. Bilcu, V. Katkovnik, and M. Vehvilainen, Color filter array interpolation based on spatial adaptivity, Proc IS&T/SPIE Electronic Imaging 2007, Computational Imaging IV, Vol. 6497, 649707, San Jose, CA, January 2007a.

D. Paliy, M. Trimeche, V. Katkovnik, and S. Alenius, Demosaicing of noisy data: Spatially adaptive approach, Proc IS&T/SPIE Electronic Imaging 2007, Computational Imaging IV, Vol. 6497, 649720, San Jose, CA, January 2007b.

K. Parulski and K.E. Spaulding, ‘‘Color image processing for digital cameras,’’ In Digital color imaging handbook, G. Sharma (Editor), CRC Press, Boca Raton, FL, 2002, pp. 728–757.

S.-C. Pei and I.-K. Tam, Effective color interpolation in CCD color filter arrays using signal correlation, IEEE Trans Circuits Syst Video Technol 13 (2003), 503–513.

R. Ramanath and W.E. Snyder, Adaptive demosaicking, J Electron Imaging 12 (2003), 633–642.

D. Taubman, Generalized Weiner reconstruction of images from colour sensor data using a scale invariant prior, Proc IEEE Int Conf Image Process 3 (2000), 801–804.

H.J. Trussel, E. Saber, and M. Vrhel, Color image processing: Vector filtering for color imaging, IEEE Signal Process Mag 22 (2005), 14–22.

M. Vega, R. Molina, and A.K. Katsaggelos, Bayesian reconstruction of color images acquired with a single CCD, Lecture Notes in Computer Science 3522 (2005), 343–350.

X. Wu and N. Zhang, Primary-consistent soft-decision color demosaicking for digital cameras (patent pending), IEEE Trans Image Process 13 (2004), 1263–1274.

L. Zhang and X. Wu, Color demosaicking via directional linear minimum mean square-error estimation, IEEE Trans Image Process 14 (2005), 2167–2178.

X. Zhang, D.A. Silverstein, J.E. Farrel, and B.A. Wandell, Color image quality metric S-CIELAB and its application on halftone texture visibility, Proc COMPCON Dig Papers, San Jose, CA, February 23–26, 1997