

# Optical Engineering

OpticalEngineering.SPIEDigitalLibrary.org

## Bayesian method application for color demosaicking

Jin Wang  
Jiaji Wu  
Zhensen Wu  
Marco Anisetti  
Gwanggil Jeon

# Bayesian method application for color demosaicking

Jin Wang,<sup>a,b</sup> Jiaji Wu,<sup>c</sup> Zhensen Wu,<sup>b</sup> Marco Anisetti,<sup>d</sup> and Gwanggil Jeon<sup>a,\*</sup>

<sup>a</sup>Incheon National University, Department of Embedded Systems Engineering, Incheon, Republic of Korea

<sup>b</sup>Xidian University, School of Physics and Optoelectronic Engineering, Xi'an, Shaanxi, China

<sup>c</sup>Xidian University, School of Electronic Engineering, Xi'an, Shaanxi, China

<sup>d</sup>Università degli Studi di Milano, Dipartimento di Informatica, Crema, Italy

**Abstract.** This study presents a Bayesian approach based on a color image demosaicking algorithm. The proposed method is composed of pointwise and patchwise measurements. The estimation of the missing pixel is formulated as a maximum *a posteriori* and a minimum energy function. By utilizing Bayesian theory and some prior knowledge, the missing color information is estimated with a statistics-based approach. Under the maximum *a posteriori* and Bayesian framework, the desired target image corresponds to the optimal reconstruction given the mosaicked image. Compared with existing demosaicking methods, the proposed algorithm improves the CPSNR, S-CIELAB, FSIM, and zipper effect measurements while maintaining high efficiency. Moreover, it handles Gaussian and Poisson noisy images better than other conventional images. © 2018 Society of Photo-Optical Instrumentation Engineers (SPIE) [DOI: 10.1117/1.OE.57.5.053102]

Keywords: Bayesian approach; Taylor series; demosaicking; color channel; real-time process; maximum a posteriori.

Paper 171728 received Nov. 1, 2017; accepted for publication Apr. 4, 2018; published online May 9, 2018.

## 1 Introduction

To reduce cost, when a digital camera captures a color image, only one color channel out of three color channels is captured. The single charge-coupled device sensor estimates the amount of red, green, and blue in the pixel. This sensor is covered with a color filter array (CFA), which causes each pixel in the camera to capture only one color channel. As pixels only have single color information, the other two color components are reconstructed, and this process is called demosaicking. One of the most famous CFA patterns called Bayer CFA is shown in Fig. 1. In this pattern, each group has nine pixels, in which five are green pixels located in a quincunx manner, and the other four pixels are red and green pixels.

Based on the Bayer pattern, various demosaicking methods were proposed in the past few decades.<sup>1,2</sup> These methods were developed to acquire a full-color image by utilizing the color differences among three color planes in the spatial or frequency domains. Especially, the adaptive color plane interpolation is an early technique that uses an edge sensing interpolation.<sup>3</sup> This method uses three predictors and then chooses one of them via an edge classifier using the Laplacian second-order derivatives for the color components and the gradient term for the green component. Based on this approach, several methods were proposed. The adaptive filtering for CFA demosaicking (AFD) uses frequency analysis of Bayer CFA samples and tries to extract high-frequency information by a properly designed filter.<sup>4</sup> Based on this analysis, AFD applies a low-pass filter for the luminance of green positions to better reconstruct high-frequency information. The high-frequency information at the red/blue positions is then reconstructed by adaptive filtering of color difference components. Color interpolation using the variance of color differences (VCD) is a heuristic approach used to determine a distinct edge;<sup>5</sup> in this approach, the

missing green values are grouped into plane and edge regions. This method uses the heuristic demosaicking approach. The high-order interpolation (HOI) method is based on a Taylor series that uses a weighted median filter to select a predictor.<sup>6</sup> This method consists of two steps: (1) interpolation to determine four estimates of a missing color and (2) an edge orientation map-based classifier. In Ref. 7, the authors proposed a  $3 \times 3$  edge strength filter that provides local, orientation-free luminance transition information. The edge strength filter is useful in recognizing the areas where a constant color difference assumption may fail, thus improving the demosaicking performance. By adding fixed weights to Ref. 8, an effective demosaicking method based on the edge property (EDAEP) was proposed.<sup>9</sup> The authors used an accurate edge detecting method to general different weights to minimize false colors in the edge areas. In Ref. 10, authors proposed to combine voting-based directional interpolation (VDI) with weighting-based interpolation for a more effective interpolation without estimating the interpolation directions erroneously in a complex region. The VDI is based on a gradient-inverse weighted interpolation along the interpolation direction as determined by a voting approach. The effective demosaicking based on subband correlation (EDUSC) was proposed using a discrete wavelet transform to discern edge pixels.<sup>11</sup> The EDUSC method uses a subimage subband correlation to enable a good initial interpolation and accurate edge detection. The EDUSC method has three parts: (1) initial interpolation in each color channel, (2) missing pixel calculation from two other color channels, and (3) iterative subband synthesis for artifact reduction. In minimized-Laplacian residual interpolation (MLRI),<sup>12</sup> the authors estimated the tentative pixel values by minimizing the Laplacian energies of the residuals. Then, the authors incorporated the MLRI into the gradient-based threshold-free algorithm to restore the demosaicking results more precisely.

\*Address all correspondence to: Gwanggil Jeon, E-mail: gjeon@inu.ac.kr

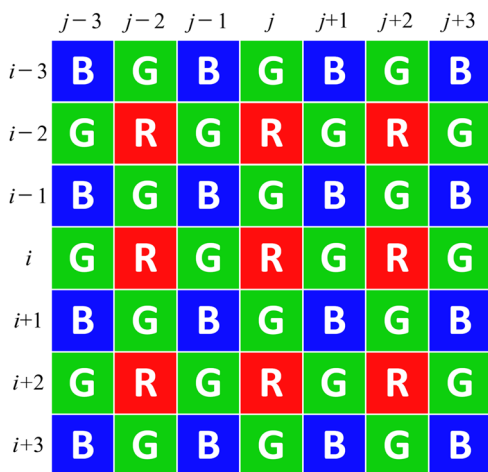


Fig. 1 7 × 7 CFA Bayer pattern.

In this study, we propose a Bayesian approach-based demosaicking algorithm (BD) with a hybrid method in consideration of pointwise and pathwise similarity. The demosaicking algorithm recovers the underlying original true color image from a mosaicked image. We formulate this method as a maximum *a posteriori* (MAP) problem. Some statistical justification should be investigated to solve the MAP problem. Therefore, we present a statistical interpretation and a Bayesian motivation to clarify the MAP issue. The MAP problem focuses on the distribution of the residual value between the missing color pixel and its neighboring color pixels. We examine the pointwise and patchwise models: the pointwise model preserves a large-scale structure, and the patchwise model provides a geometric outline and a fine-scale structure in a large search region. In the computational complexity concept, the pointwise model is simpler and more efficient than the patchwise model. To utilize the different advantages of point and patch structures, we design an adaptive hybrid model as a linear combination

of the two models, and the parameter is adaptive to a standard deviation representing the structural feature. After the interpolation of the missing green channel, we estimate the red/blue channel by utilizing the uniform color difference in a local region. Afterward, we apply demosaicking algorithms to the Gaussian and Poisson noisy images to prove the robustness of the proposed algorithm.

The rest of the paper is organized as follows: in Sec. 2, we describe the proposed algorithm. In Sec. 3, some experimental results and the corresponding discussion are provided. Finally, the conclusion is drawn in Sec. 4.

## 2 Proposed Algorithm

### 2.1 Bayesian Approach-Based Analysis

Let a mosaicked image be  $\mathbf{x}(\mathbf{k}) = [\mathbf{x}_R(\mathbf{k})\mathbf{x}_G(\mathbf{k})\mathbf{x}_B(\mathbf{k})]$ , where  $\mathbf{k} = (i, j) \in R^2$  is the image coordinate index. We assume that the true original color image  $\mathbf{y} = [\mathbf{y}_R(\mathbf{k})\mathbf{y}_G(\mathbf{k})\mathbf{y}_B(\mathbf{k})]$  is highly related to  $\mathbf{x}$ , as shown in Fig. 2. An estimator  $\hat{\mathbf{y}}$  is considered a Bayes estimator if it minimizes  $E[L(\mathbf{y}, \hat{\mathbf{y}})|\mathbf{x}]$  among all estimators. It is equivalent to an MAP solution, in which  $L(\mathbf{y}, \hat{\mathbf{y}})$  is a loss function used as some function of the difference between estimated and true values. The quadratic loss function  $L(\mathbf{y}, \hat{\mathbf{y}}) = \|\hat{\mathbf{y}} - \mathbf{y}\|^2$  is commonly used in signal processing, and  $\lambda(\mathbf{y})$  is a constant. The optimal Bayesian estimator (OBE)  $\hat{\mathbf{y}}_{\text{OBE}}$  is determined by minimizing  $E[L(\mathbf{y}, \hat{\mathbf{y}})|\mathbf{x}]$  as

$$\hat{\mathbf{y}}_{\text{OBE}} = \arg \min_{\hat{\mathbf{y}}} E[\|\hat{\mathbf{y}} - \mathbf{y}\|^2|\mathbf{x}]. \tag{1}$$

Based on the assumption of the quadratic loss function,<sup>13</sup> the OBE is

$$\hat{\mathbf{y}}_{\text{OBE}} = E[\mathbf{y}|\mathbf{x}] = \sum_{\mathbf{y} \in \Lambda} \mathbf{y} p(\mathbf{y}|\mathbf{x}) = \frac{\sum_{\mathbf{y} \in \Lambda} \mathbf{y} p(\mathbf{x}|\mathbf{y}) p(\mathbf{y})}{p(\mathbf{x})}, \tag{2}$$

where  $\Lambda$  is the large space of all configurations of  $\mathbf{y}$  ( $|\Lambda| = 256^n$  if  $y(x) \in \{0, \dots, 255\}$ ). Here parameter  $n$  is

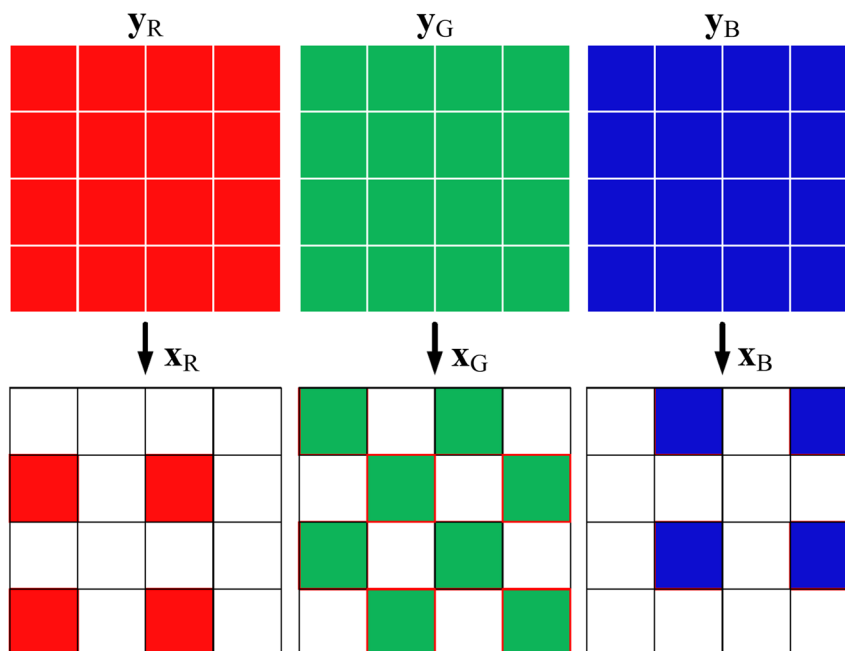


Fig. 2 Observation of the low resolution for the Bayer CFA.

the size of the data. By using the Bayes rules and the law of total probability, we have

$$p(\mathbf{x}, \mathbf{y}) = p(\mathbf{x}|\mathbf{y})p(\mathbf{y}), \quad p(\mathbf{x}) = \sum_{\mathbf{y} \in \Lambda} p(\mathbf{x}|\mathbf{y})p(\mathbf{y}).$$

Then, we can modify Eq. (2) into

$$\hat{\mathbf{y}}_{\text{OBE}} = \frac{\sum_{\mathbf{y} \in \Lambda} \mathbf{y} p(\mathbf{x}|\mathbf{y})p(\mathbf{y})}{\sum_{\mathbf{y} \in \Lambda} p(\mathbf{x}|\mathbf{y})p(\mathbf{y})}. \quad (3)$$

As shown in Eq. (3), to obtain the optimal estimation  $\hat{\mathbf{y}}_{\text{OBE}}$ , we determine the distribution of  $\mathbf{x}|\mathbf{y}$  and the distribution of  $\mathbf{y}$  known as a *prior* distribution. To derive the distributions, we need to utilize the observation in a local neighbor region  $S_y$  centered at the to-be-interpolated pixel instead of the pixels at the same position. In the local region  $S_y$ , we assume that the prior distribution of  $\mathbf{y}$  is a uniform

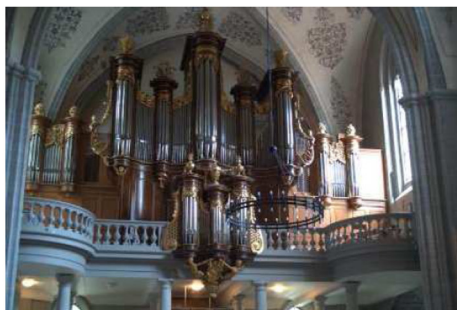
distribution, that is,  $p(\mathbf{y}) = 1/|S_y|$ . In this local neighbor region, we assume that the observation has similar structures. To achieve the optimal estimation, we use the following approximation:<sup>13</sup>

$$\frac{1}{|S_y|} \sum_{\mathbf{m} \in S_y} \mathbf{y}(\mathbf{k} + \mathbf{m}) p[\mathbf{x}(\mathbf{k})|\mathbf{y}(\mathbf{k} + \mathbf{m})] \rightarrow \sum_{\mathbf{y} \in \Lambda} \mathbf{y} p(\mathbf{x}|\mathbf{y})p(\mathbf{y}),$$

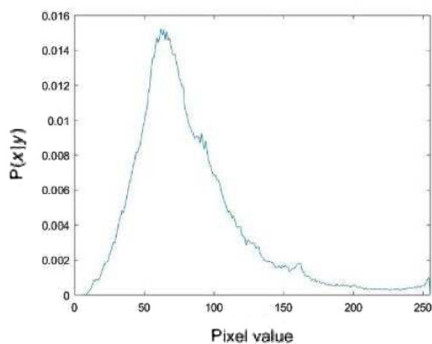
$$\frac{1}{|S_y|} \sum_{\mathbf{m} \in S_y} p[\mathbf{x}(\mathbf{k})|\mathbf{y}(\mathbf{k} + \mathbf{m})] \rightarrow \sum_{\mathbf{y} \in \Lambda} p(\mathbf{x}|\mathbf{y})p(\mathbf{y}). \quad (4)$$

Then, Eq. (3) is changed into

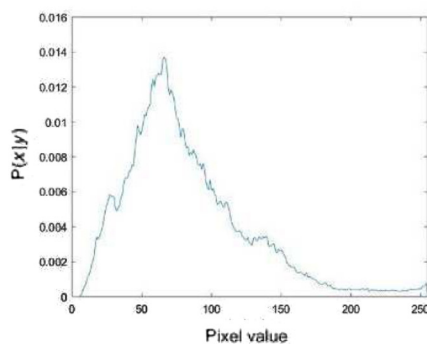
$$\hat{\mathbf{y}}_{\text{OBE}} \approx \frac{\frac{1}{|S_y|} \sum_{\mathbf{m} \in S_y} \mathbf{y}(\mathbf{k} + \mathbf{m}) p[\mathbf{x}(\mathbf{k})|\mathbf{y}(\mathbf{k} + \mathbf{m})]}{\frac{1}{|S_y|} \sum_{\mathbf{m} \in S_y} p[\mathbf{x}(\mathbf{k})|\mathbf{y}(\mathbf{k} + \mathbf{m})]}, \quad (5)$$



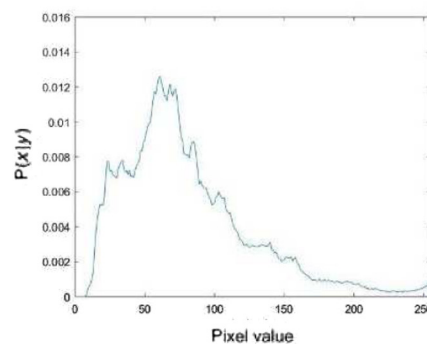
(a)



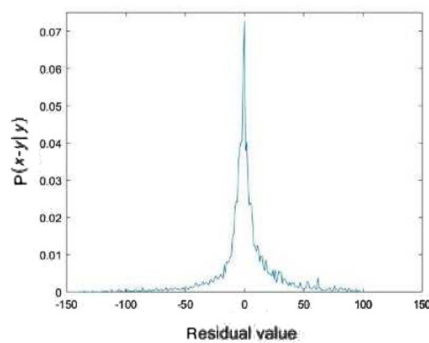
(b)



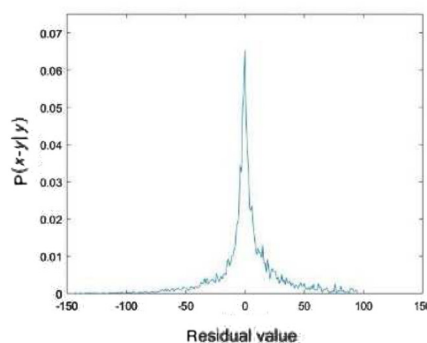
(c)



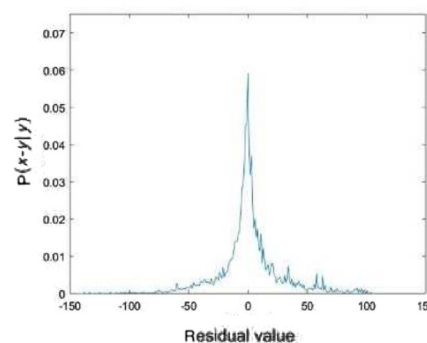
(d)



(e)



(f)



(g)

**Fig. 3** Illustration of the image statistic model. (a) Original image, (b)  $p[\mathbf{x}(\mathbf{k})|\mathbf{y}(\mathbf{k} + \mathbf{m})]$  in the R color channel, (c)  $p[\mathbf{x}(\mathbf{k})|\mathbf{y}(\mathbf{k} + \mathbf{m})]$  in the G color channel, (d)  $p[\mathbf{x}(\mathbf{k})|\mathbf{y}(\mathbf{k} + \mathbf{m})]$  in the B color channel, (e)  $p[\mathbf{x}(\mathbf{k}) - \mathbf{y}(\mathbf{k} + \mathbf{m})|\mathbf{y}(\mathbf{k} + \mathbf{m})]$  in the R color channel, (f)  $p[\mathbf{x}(\mathbf{k}) - \mathbf{y}(\mathbf{k} + \mathbf{m})|\mathbf{y}(\mathbf{k} + \mathbf{m})]$  in the G color channel, and (g)  $p[\mathbf{x}(\mathbf{k}) - \mathbf{y}(\mathbf{k} + \mathbf{m})|\mathbf{y}(\mathbf{k} + \mathbf{m})]$  in the B color channel.

where  $\mathbf{k} + \mathbf{m}$  can be any neighbor pixels in a square region  $(2w_s + 1) \times (2w_s + 1)$ . In this paper, we choose  $w_s = 1$  for high efficiency. The target is changed to find the distribution of  $\mathbf{x}(\mathbf{k})|\mathbf{y}(\mathbf{k} + \mathbf{m})$ . Figures 3(b)–3(d) show that it does not have the obvious statistical property for all the three color channels. Based on the histogram of residual images, we derive the distribution of  $\mathbf{x}(\mathbf{k}) - \mathbf{y}(\mathbf{k} + \mathbf{m})|\mathbf{y}(\mathbf{k} + \mathbf{m})$  instead of  $\mathbf{x}(\mathbf{k})|\mathbf{y}(\mathbf{k} + \mathbf{m})$ . Figures 3(e)–3(g) indicate that  $\mathbf{x}(\mathbf{k}) - \mathbf{y}(\mathbf{k} + \mathbf{m})|\mathbf{y}(\mathbf{k} + \mathbf{m})$  has a clear statistical feature. Two well-known distributions, namely, Gaussian and Laplacian, have this kind of form. We compare these two kinds of distributions with different parameters in Fig. 4. The Laplacian distribution fits the statistical property better than the Gaussian distribution. The distribution of  $\mathbf{x}(\mathbf{k})|\mathbf{y}(\mathbf{k} + \mathbf{m})$  follows the Laplacian distribution.

Equation (5) is modified into

$$\hat{\mathbf{y}}_G(\mathbf{k}) = \frac{\sum_{\mathbf{m} \in S_y} \tilde{\mathbf{y}}_G(\mathbf{k} + \mathbf{m}) C(\mathbf{y})}{\sum_{\mathbf{m} \in S_y} C(\mathbf{y})}, \quad (6)$$

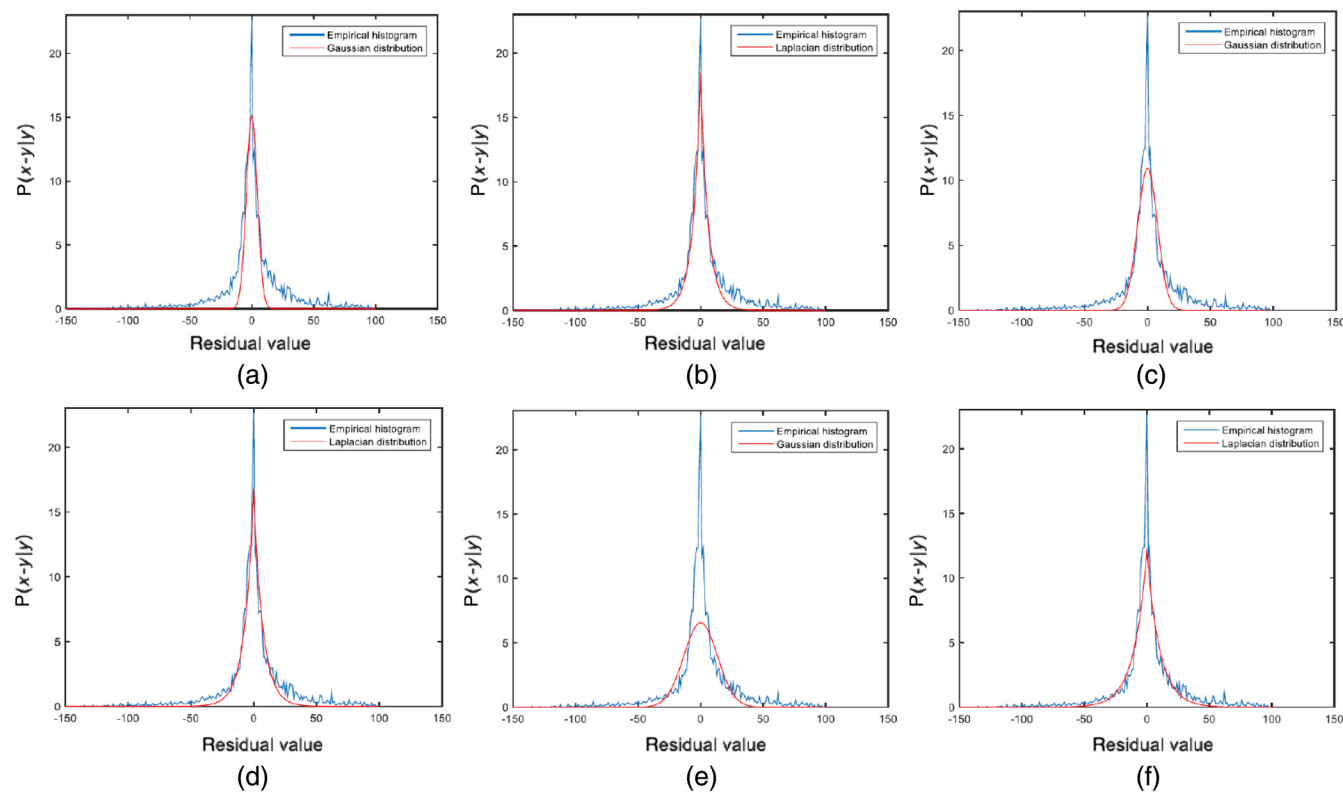
where  $C(\mathbf{y}) = \exp[-\|\mathbf{v}_G(\mathbf{k}) - \mathbf{v}_G(\mathbf{k} + \mathbf{m})\|_1 / \alpha]$ .

The weight parameter  $\alpha$  plays an important role in the statistic model. For a different weight parameter  $\alpha$ , the matching has varying features, as shown on the left in Fig. 4. In the figure, we depict the empirical histogram of residual images and compare the empirical histogram with the prior statistical model of varying parameters. From the top-left to the bottom-left, the weight parameter  $\alpha$  is 0.35, 0.36, and 0.4. The top one represents the parameters with the smallest  $\alpha$ ,

which achieves good matching at zero but fails at tails. The bottom one represents the parameters with the largest  $\alpha$ , which achieves good matching at tails but fails at zero. In Fig. 4, the width plays a more important role than height because when the residual is zero, this pixel is in the smooth region, which is easy to be reconstructed with conventional methods. In terms of image restoration, the focus is on the complex region such as edges or textures, that is, the tails of the distribution. The tails are controlled by the width. To achieve the best matching with the empirical histogram, we should assign adaptive weight parameters to different pixel values. That is, large residual values should have large weight parameters, and small residual values should have small ones. Therefore, we design an adaptive weight parameter that is proportional to the variance of the to-be-interpolated pixel in a local region

$$\alpha = \lambda \sigma_m = \sum_{\mathbf{m} \in S_y} \frac{\lambda \|\mathbf{x}(\mathbf{k} + \mathbf{m}) - \bar{\mathbf{x}}\|}{|S_y|}, \quad (7)$$

where  $\bar{\mathbf{x}}$  is the expected value in the local region  $S_y$ , with size  $(2w_s + 1) \times (2w_s + 1)$ . Through Eq. (6), we can determine the original image  $\mathbf{y}(\mathbf{k} + \mathbf{m})$  to obtain the optimal estimation of the color image. The original image is partially unknown, but it is also partially given in the mosaicked image. Thus, in Eq. (6), we substitute the unknown  $\mathbf{y}(\mathbf{k} + \mathbf{m})$  into the estimated direction value of the given mosaicked color pixel values.



**Fig. 4** Matching of the statistical model and empirical histogram with different parameters of the Gaussian and Laplacian distributions. (a)  $f(x) = 3.8 \times 10^{-5} \exp(-0.026x^2)$ , (b)  $f(x) = 4.6 \times 10^{-5} \exp(-0.16|x|)$ , (c)  $f(x) = 2.8 \times 10^{-5} \exp(-0.0081x^2)$ , (d)  $f(x) = 4.2 \times 10^{-5} \exp(-0.137|x|)$ , (e)  $f(x) = 1.7 \times 10^{-5} \exp(-0.0028x^2)$ , and (f)  $f(x) = 3 \times 10^{-5} \exp(-0.096|x|)$ .

## 2.2 Green Channel Reconstruction Using a Bayesian Framework

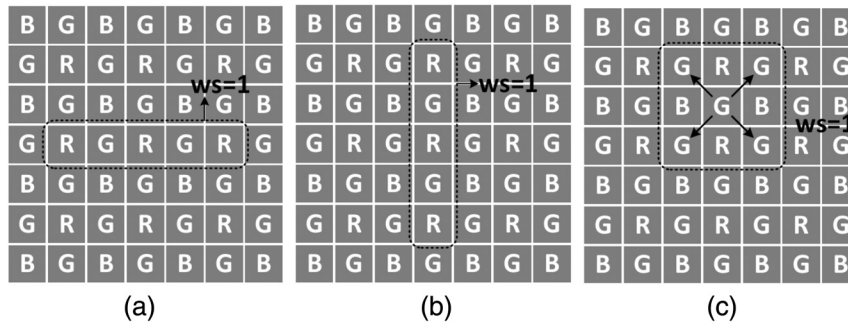
According to the discussion in Sec. 2.1, we first restore the green channel and then reconstruct the red/blue channel based on the entire green channel and the given mosaicking color pixels. Figure 1 shows that a missing green color pixel at a given red/blue color pixel is surrounded by the four nearest given green color pixels in the north, south, west, and east directions. We still have the red/blue color pixel in the missing green color pixel. Therefore, we should consider this feature to reconstruct the missing green color pixel. The mosaicked image has a special feature that has pixel values in all locations. Based on this specific property of mosaicking image, instead of directly using the green color pixels in the four directions in Eq. (6), we adopt the Taylor approximation<sup>14</sup> to provide the approximated four-direction color values by the given color pixel values

$$\mathbf{y}_G(\mathbf{k} + \mathbf{m}) = \mathbf{x}_G(\mathbf{k} + \mathbf{m}) + \frac{\mathbf{x}_R(\mathbf{k}) - \mathbf{x}_R(\mathbf{k} + 2\mathbf{m})}{2} + \frac{\mathbf{x}_R(\mathbf{k} + 3\mathbf{m}) + \mathbf{x}_R(\mathbf{k} - \mathbf{m}) - 2\mathbf{x}_R(\mathbf{k} + \mathbf{m})}{8}. \quad (8)$$

In Eq. (6), we have two candidates for distance  $\|\mathbf{x}(\mathbf{k}) - \mathbf{y}(\mathbf{k} + \mathbf{m})\|$ , namely, pointwise and patchwise. The patchwise distance aims to preserve the structure of the images in a large local region. In a small search region, especially in terms of image restoration, pointwise similarity plays an important role in addressing a large-scale structure. Both pointwise and patchwise distances should be considered to estimate the missing color pixels. Therefore, we propose the following hybrid Bayesian estimator (HBE)-BD algorithm:

$$C(\mathbf{y}) = \exp\left[\frac{-\|\mathbf{v}_G(\mathbf{k}) - \mathbf{v}_G(\mathbf{k} + \mathbf{m})\|_1}{2\alpha}\right] + \mu \exp\left[\frac{-\|\mathbf{x}_R(\mathbf{k}) - \mathbf{x}_R(\mathbf{k} + \mathbf{m})\|_1}{\alpha}\right], \quad (9)$$

where  $\mathbf{v}_G(\mathbf{k})$  and  $\mathbf{v}_G(\mathbf{k} + \mathbf{m})$  are the patch vectors containing the given center red pixel and four neighboring green pixels. The neighboring region  $S_y$  is shown in Fig. 5. In this algorithm,  $\mu = 0.1$ .



**Fig. 5** Structure of the neighboring region. (a)  $S_y^H$  is the region containing the given color pixels in the horizontal direction and with size  $1 \times (2ws + 1)$ , (b)  $S_y^V$  is the region containing the given color pixels in the vertical direction and with size  $(2ws + 1) \times 1$ , and (c)  $S_y^{RB}$  contains the four given red pixels with size  $(2ws + 1) \times (2ws + 1)$ .

For images with complex features, such as various directions or textures, Eq. (6) can obtain good performance. For the two special cases of horizontal and vertical edges, we should apply the given pixels in the horizontal or vertical direction to reconstruct the missing color pixel. Horizontal and vertical directions exist in most images and occupy a large proportion in one image. Therefore, we estimate the two directions by modifying Eq. (6) and adopting the horizontal and vertical components as follows:

$$\hat{\mathbf{y}}_G^H(\mathbf{k}) = \frac{\sum_{\mathbf{m} \in S_y^H} \tilde{\mathbf{y}}_G(\mathbf{k} + \mathbf{m}) C(\mathbf{y})}{\sum_{\mathbf{m} \in S_y^H} C(\mathbf{y})},$$

$$\hat{\mathbf{y}}_G^V(\mathbf{k}) = \frac{\sum_{\mathbf{m} \in S_y^V} \tilde{\mathbf{y}}_G(\mathbf{k} + \mathbf{m}) C(\mathbf{y})}{\sum_{\mathbf{m} \in S_y^V} C(\mathbf{y})}, \quad (10)$$

where  $S_y^H$  and  $S_y^V$  are the regions containing the given color pixels in the horizontal and vertical directions and with size  $1 \times (2ws + 1)$  and  $(2ws + 1) \times 1$  (Fig. 5), respectively. Here,  $C(\mathbf{y})$  is adopted in Eq. (9). Then, a criterion is designed to distinguish the direction as follows:

$$\bar{\mathbf{y}}_G(\mathbf{k}) = \begin{cases} \hat{\mathbf{y}}_G^H(\mathbf{k}), & \text{if } \delta(\mathbf{k}) > T_1 \\ \hat{\mathbf{y}}_G^V(\mathbf{k}), & \text{if } \delta(\mathbf{k}) < -T_1, \\ \hat{\mathbf{y}}_G(\mathbf{k}), & \text{otherwise} \end{cases}$$

$$\delta(\mathbf{k}) = \begin{cases} 1, & \text{if } \min(\mathbf{D}_H(\mathbf{k}), \mathbf{D}_V(\mathbf{k}), T_2) = \mathbf{D}_H(\mathbf{k}) \\ -1, & \text{if } \min(\mathbf{D}_H(\mathbf{k}), \mathbf{D}_V(\mathbf{k}), T_2) = \mathbf{D}_V(\mathbf{k}), \\ 0, & \text{otherwise} \end{cases}$$

$$\mathbf{D}_H = \mathbf{x} * \mathbf{f} + (\mathbf{x} - \hat{\mathbf{y}}_G^H) * \mathbf{f},$$

$$\mathbf{D}_V = \mathbf{x} * \mathbf{f}' + (\mathbf{x} - \hat{\mathbf{y}}_G^V) * \mathbf{f}', \quad (11)$$

where  $T$  is the threshold, and  $\mathbf{D}_H$  and  $\mathbf{D}_V$  are normalized. The interpolated full green channel is then obtained.

## 2.3 Red/Blue Plane Reconstruction

As we fully obtained a green plane in Sec. 2.2, we now populate red/blue pixels in blue/red or green locations. In the reconstruction of the red/blue color channel, we use the restored green color channel because of the strong correlation among the color channels. To reconstruct the missing

red pixels in the given blue positions, four given red pixels are located around the given blue pixel (Fig. 1). We use the color difference based on the reconstructed green pixels as follows:

$$\begin{aligned}\bar{y}_R(\mathbf{k}) &= \bar{y}_G(\mathbf{k}) + \frac{\sum_{\mathbf{m} \in S_y^{RB}} \Delta(\mathbf{k} + \mathbf{m}) C(\mathbf{y})}{\sum_{\mathbf{m} \in S_y^{RB}} C(\mathbf{y})}, \\ \Delta(\mathbf{k} + \mathbf{m}) &= \mathbf{x}_R(\mathbf{k} + \mathbf{m}) - \bar{y}_G(\mathbf{k} + \mathbf{m}), \\ C(\mathbf{y}) &= \exp \left[ \frac{-\|\mathbf{v}_R(\mathbf{k}) - \mathbf{v}_R(\mathbf{k} + \mathbf{m})\|_1}{\beta} \right], \\ &+ \mu \exp \left[ \frac{-\|\bar{y}_G(\mathbf{k}) - \bar{y}_R(\mathbf{k} + \mathbf{m})\|_1}{\beta} \right], \\ \beta &= \frac{\alpha}{2},\end{aligned}\quad (12)$$

where  $\mathbf{v}_R(\mathbf{k})$  and  $\mathbf{v}_R(\mathbf{k} + \mathbf{m})$  are the patch vectors including the four given red pixels located in the northwest, northeast, southwest, and southeast, and  $S_y^{RB}$  contains the same four given red pixels with size  $(2ws + 1) \times (2ws + 1)$  as shown in Fig. 5(c).

For the missing red/blue pixel in the given  $G$  pixel, Eq. (13) is applied as shown in Fig. 1.

$$\begin{aligned}\bar{y}_R(\mathbf{k}) &= \bar{y}_G(\mathbf{k}) + \frac{\sum_{\mathbf{m} \in S_y^{RG}} \Delta(\mathbf{k} + \mathbf{m}) C(\mathbf{y})}{\sum_{\mathbf{m} \in S_y^{RG}} C(\mathbf{y})}, \\ \Delta(\mathbf{k} + \mathbf{m}) &= \mathbf{x}_R(\mathbf{k} + \mathbf{m}) - \bar{y}_G(\mathbf{k} + \mathbf{m}), \\ C(\mathbf{y}) &= \exp \left[ \frac{-\|\bar{y}_G(\mathbf{k}) - \bar{y}_R(\mathbf{k} + \mathbf{m})\|_1}{\beta} \right],\end{aligned}\quad (13)$$

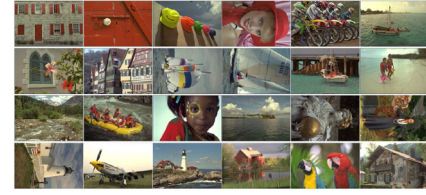
where  $S_y^{RG}$  contains the two given red pixels above and below the given green pixel.

## 2.4 Refinement of Each Color Channel

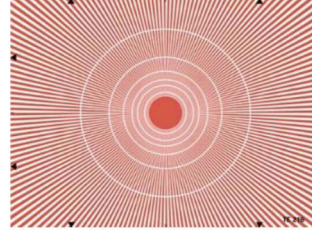
Once the green channel is populated using Eq. (11), for the given red pixels, the green channel can be further refined as

$$\begin{aligned}\bar{y}_R(\mathbf{k}) &= \mathbf{x}_R(\mathbf{k}) + \kappa [\hat{y}_G(\mathbf{k}) - \mathbf{x}_R(\mathbf{k})] \\ &+ (1 - \kappa) \frac{\sum_{\mathbf{m} \in S(x)} \Delta(\mathbf{k} + \mathbf{m}) \hat{x}_G(\mathbf{k} + \mathbf{m})}{\sum_{\mathbf{m} \in S(x)} \Delta(\mathbf{k} + \mathbf{m})}, \\ \hat{x}_G(\mathbf{k} + \mathbf{m}) &= \mathbf{x}_G(\mathbf{k} + \mathbf{m}) - \bar{x}_R(\mathbf{k} + \mathbf{m}), \\ \Delta(\mathbf{k} + \mathbf{m}) &= \left\{ |\hat{y}_G(\mathbf{k}) - \mathbf{x}_G(\mathbf{k} + \mathbf{m})| \right. \\ &\left. + \sum_{a \in \{0,2\}} |\mathbf{x}_R(\mathbf{k} + a\mathbf{m}) - \mathbf{x}_R[\mathbf{k} + (a+2)\mathbf{m}]| \right\}^{-1},\end{aligned}\quad (14)$$

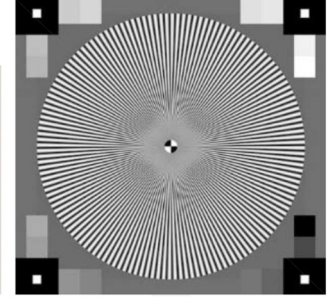
where  $\kappa$  is the weight parameter assumed to be 0.1. Then, we refine the red/blue channel by the modified green channel as in Sec. 2.2. We iteratively refine the green channel and red/blue channel by using the above procedure until we obtain the optimal results or reach the stop condition.



(a)



(b)



(c)

**Fig. 6** (a) 24 images of the Kodak dataset, (b) high-frequency image TE216, and (c) high-frequency image TE253.

## 3 Experimental Results

This section presents the detailed performance comparison. Simulations were conducted on  $512 \times 768$  size Kodak dataset<sup>15</sup> and two high-frequency images<sup>16</sup> to assess the superiority of the proposed method compared with the conventional methods. The test images are shown in Fig. 6. Eight methods were selected as benchmarks: AFD,<sup>4</sup> VCD,<sup>5</sup> HOI,<sup>6</sup> ESF,<sup>7</sup> EDAEP,<sup>8</sup> VDI,<sup>10</sup> EDUSC,<sup>11</sup> and MLRI.<sup>12</sup> We evaluated the objective performance in terms of CPSNR, S-CIELAB  $\Delta E^*$ , FSIM, and the zipper effect (ZE) aside from the visual performance. Afterward, we apply the Gaussian and Poisson noise to the Kodak dataset and provide the CPSNR results through a comparison with the existing demosaicking methods. Our simulation was conducted on an Intel Core 2 Duo CPU E8500 @ 3.16 GHz. The window size  $ws$  was set to 2.

### 3.1 Objective Performance Analysis

We first compare the objective performances measured in terms of CPSNR. Table 1 displays the comparison of the CPSNR results of the presented method with those of other benchmark methods. Seven methods were selected as benchmarks: AFD, VCD, HOI, ESF, EDAEP, VDI, and MLRI. As shown in Table 1, the proposed method has the best average CPSNR, and the gain are 1.229 (AFD), 0.916 (VCD), 1.473 (HOI), 0.519 (ESF), 2.786 (EDAEP), 2.699 (VDI), and 1.233 (MLRI). Although images 4, 15, 16, 23, and 24 did not show the best CPSNR using the proposed method, our method still ranks second. ‘‘Diff.’’ shows the CPSNR difference between BD and the corresponding method. ‘‘Rank’’ indicates the rank of the BD method compared with those of the benchmark methods. The best performances are marked in bold.

To numerically evaluate the results of visual performance, we used the S-CIELAB  $\Delta E^*$  metric.<sup>17</sup> S-CIELAB  $\Delta E^*$  metric is a spatial form of the CIELAB  $\Delta E^*$  developed for determining the distance between the S-CIELAB  $\Delta E^*$  representation of an original and that of the reconstructed image.

**Table 1** CPSNR performance comparison on the Kodak images (the best performances are marked in bold).

Image	AFD	VCD	HOI	ESF	EDAEP	VDI	MLRI	BD	Rank
1	37.558	38.419	36.564	39.909	34.919	35.343	36.805	<b>40.676</b>	1
2	40.7	40.433	40.827	40.821	39.736	39.607	40.791	<b>41.565</b>	1
3	42.686	42.537	42.71	42.554	41.749	41.318	42.987	<b>43.517</b>	1
4	41.055	40.769	40.716	40.447	40.034	39.56	<b>41.49</b>	41.363	2
5	38.039	37.783	37.634	37.55	35.782	36.542	37.657	<b>38.68</b>	1
6	38.036	39.999	38.807	41.217	36.845	36.98	39.131	<b>41.386</b>	1
7	42.907	42.133	42.722	42.154	41.281	41.828	42.841	<b>43.223</b>	1
8	35.23	36.289	34.855	37.189	32.832	33.995	34.894	<b>37.631</b>	1
9	42.57	42.85	42.391	42.959	40.902	41.687	42.377	<b>43.439</b>	1
10	42.699	42.465	42.268	42.602	40.978	41.162	42.419	<b>43.154</b>	1
11	39.338	39.889	39.128	40.693	37.476	37.685	39.305	<b>40.926</b>	1
12	42.774	43.43	43.027	43.801	41.669	41.705	43.268	<b>44.277</b>	1
13	33.769	34.798	32.767	36.106	31.484	31.233	33.141	<b>36.326</b>	1
14	37.166	37.012	37.173	36.113	36.268	36.336	37.615	<b>37.898</b>	1
15	<b>39.839</b>	39.695	39.054	39.274	38.805	38.071	39.339	39.808	2
16	41.148	43.713	42.467	<b>44.772</b>	40.441	40.65	42.879	44.289	2
17	41.392	41.519	40.852	41.953	39.631	39.639	40.985	<b>42.107</b>	1
18	37.118	37.116	36.449	37.72	35.379	34.967	36.585	<b>37.86</b>	1
19	40	40.506	39.82	41.494	37.752	38.869	39.954	<b>41.684</b>	1
20	41.117	41.128	40.629	41.523	39.26	39.744	40.681	<b>41.752</b>	1
21	38.676	39.16	37.915	40.297	36.452	36.971	38.181	<b>40.375</b>	1
22	38.512	38.072	38.031	38.409	37.173	37.123	38.614	<b>38.965</b>	1
23	43.149	42.995	43.483	42.449	42.356	41.309	<b>43.81</b>	43.726	2
24	34.832	35.103	34.162	<b>35.342</b>	33.728	32.708	34.704	35.177	2
Avg.	39.596	39.909	39.352	40.306	38.039	38.126	39.602	<b>40.825</b>	1
Diff.	1.229	0.916	1.473	0.519	2.786	2.699	1.223		

**Table 2** S-CIELAB  $\Delta E^*$  results on Kodak dataset for several demosaicking methods (the best performances are marked in bold).

Image	AFD	VCD	HOI	ESF	EDAEP	VDI	MLRI	BD	Rank
1	1.1989	1.1246	1.2505	0.9954	1.4736	1.368	1.2032	<b>0.9648</b>	1
2	0.6427	0.6818	0.6442	0.6634	0.7029	0.7283	0.6615	<b>0.6061</b>	1
3	0.4889	0.501	0.4916	0.5095	0.5243	0.5413	0.4706	<b>0.4632</b>	1
4	0.6721	0.6907	0.7003	0.7266	0.7285	0.7953	<b>0.6406</b>	0.6595	2
5	<b>1.0136</b>	1.0561	1.0641	1.2419	1.2706	1.1074	1.1107	1.0406	2



**Table 2** (Continued).

Image	AFD	VCD	HOI	ESF	EDAEP	VDI	MLRI	BD	Rank
6	0.9571	0.8125	0.8515	<b>0.7413</b>	0.9889	0.9498	0.8347	0.755	2
7	0.5229	0.5942	0.5277	0.5838	0.6637	0.5641	0.5364	<b>0.5081</b>	1
8	1.4127	1.2863	1.4158	1.1904	1.8245	1.4773	1.4597	<b>1.1727</b>	1
9	0.582	0.5747	0.6007	0.5852	0.6529	0.6238	0.5772	<b>0.5485</b>	1
10	0.5367	0.5563	0.5715	0.569	0.6102	0.617	0.5554	<b>0.5327</b>	1
11	0.8032	0.763	0.7803	0.7223	0.9226	0.8587	0.7695	<b>0.7097</b>	1
12	0.5062	0.4779	0.4837	0.4655	0.5501	0.529	0.4794	<b>0.4405</b>	1
13	1.6959	1.6411	1.858	<b>1.4771</b>	2.0124	2.0417	1.8216	1.5482	2
14	1.0179	0.995	0.9873	1.0433	1.1376	1.0914	0.9364	<b>0.8894</b>	1
15	<b>0.6423</b>	0.6712	0.7007	0.7215	0.6907	0.7257	0.6829	0.6673	2
16	0.6983	0.5556	0.5979	<b>0.508</b>	0.7042	0.6892	0.5628	0.5828	4
17	0.5419	0.5529	0.564	0.5426	0.6409	0.6175	0.5713	<b>0.5365</b>	1
18	<b>1.0077</b>	1.0575	1.0694	1.0919	1.16	1.1466	1.0518	1.0563	3
19	0.7821	0.7746	0.8036	0.7357	0.9565	0.8466	0.7853	<b>0.7343</b>	1
20	0.5459	0.5621	0.5764	0.5553	0.6474	0.5785	0.675	<b>0.5458</b>	1
21	0.9291	0.9172	0.9886	<b>0.8606</b>	1.0905	1.0189	0.962	0.8917	2
22	0.9421	1.0344	0.9805	1.0229	1.0563	0.9952	<b>0.914</b>	0.9243	2
23	0.5009	0.528	0.4877	0.5596	0.5359	0.5439	<b>0.4709</b>	0.4872	2
24	<b>1.0181</b>	1.0775	1.0992	1.1017	1.1561	1.1183	1.143	1.1416	6
Avg.	0.8191	0.8119	0.8373	0.8006	0.9459	0.8989	0.8282	<b>0.767</b>	1
Diff.	0.0522	0.045	0.0703	0.0337	0.1789	0.1319	0.0612		

**Table 3** FSIM results on the Kodak dataset for several demosaicking methods (the best performances are marked in bold).

Image	AFD	VCD	HOI	ESF	EDAEP	VDI	MLRI	BD	Rank
1	0.9986	0.9987	0.998	0.999	0.9974	0.9975	0.9983	<b>0.9992</b>	1
2	0.9985	0.9984	0.9982	0.9984	0.9981	0.9976	0.9983	<b>0.9986</b>	1
3	0.9993	0.9992	0.9992	0.9992	0.9991	0.9986	0.9993	<b>0.9994</b>	1
4	0.9992	0.9992	0.9989	0.9992	0.9989	0.9986	0.9992	<b>0.9992</b>	1
5	0.9989	0.9988	0.9988	0.9988	0.9982	0.9985	0.9988	<b>0.9991</b>	1
6	0.9987	0.999	0.9987	0.9993	0.9982	0.9982	0.9989	<b>0.9993</b>	1
7	0.9995	0.9995	0.9995	0.9995	0.9993	0.9993	0.9995	<b>0.9996</b>	1
8	0.9978	0.9982	0.9975	0.9985	0.9963	0.9972	0.9977	<b>0.9987</b>	1
9	0.9992	0.9992	0.999	0.9993	0.9988	0.9988	0.9991	<b>0.9993</b>	1
10	0.9993	0.9992	0.9991	0.9993	0.9989	0.9988	0.9992	<b>0.9993</b>	1

**Table 3** (Continued).

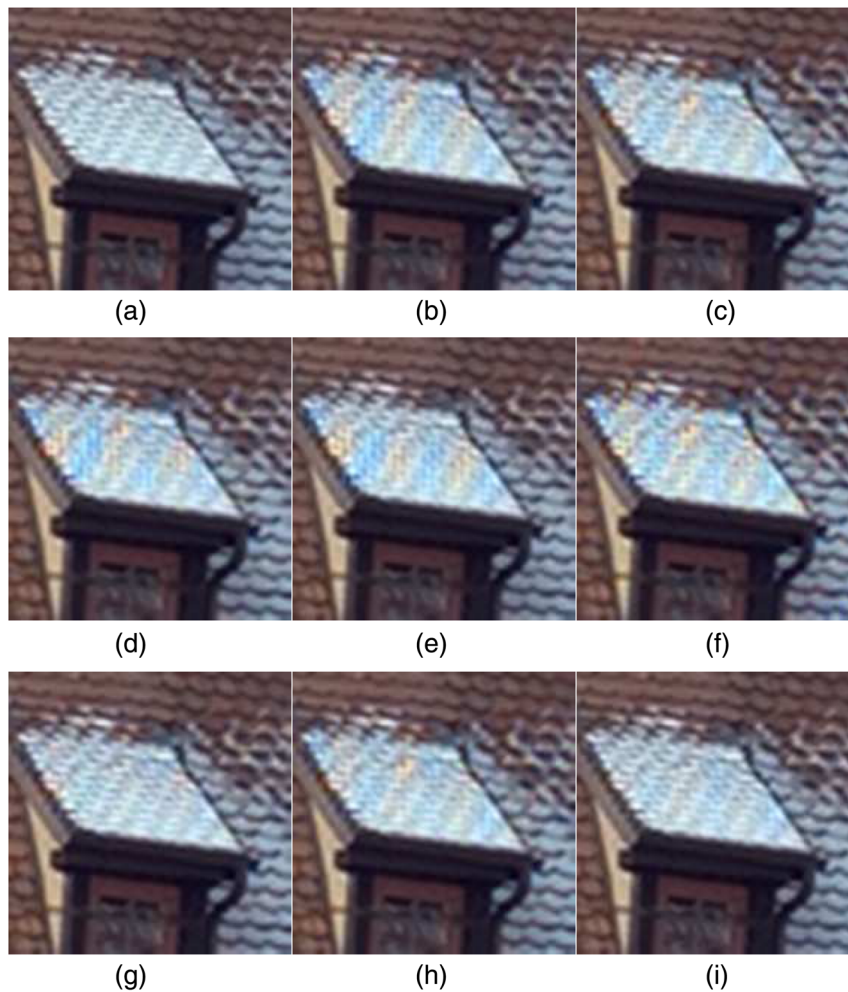
Image	AFD	VCD	HOI	ESF	EDAEP	VDI	MLRI	BD	Rank
11	0.9988	0.9989	0.9986	0.9991	0.9981	0.9981	0.9988	<b>0.9991</b>	1
12	0.9991	0.9991	0.999	0.9993	0.9987	0.9987	0.9991	<b>0.9993</b>	1
13	0.9974	0.9978	0.9962	0.9985	0.9954	0.9953	0.9969	<b>0.9987</b>	1
14	0.9987	0.9987	0.9985	0.9985	0.9983	0.9979	0.9987	<b>0.9989</b>	1
15	0.9986	0.9987	0.9978	0.9985	0.998	<b>0.9972</b>	0.9983	0.9984	4
16	0.999	0.9994	0.9991	0.9995	0.9988	0.9988	<b>0.9993</b>	0.9994	2
17	0.9993	0.9993	0.9991	0.9994	0.9989	0.999	0.9993	<b>0.9994</b>	1
18	0.9983	0.9982	0.9977	0.9986	0.9973	0.997	0.9978	<b>0.9986</b>	1
19	0.999	0.9991	0.9987	0.9992	0.9984	0.9986	0.999	<b>0.9993</b>	1
20	0.9991	0.999	0.9989	0.9991	0.9987	0.9987	0.9991	<b>0.9992</b>	1
21	0.9985	0.9986	0.9981	0.9989	0.9976	0.9978	0.9984	<b>0.9989</b>	1
22	0.9984	0.9982	0.9981	0.9983	0.9978	0.9978	0.9985	<b>0.9985</b>	1
23	0.9993	0.9993	0.9993	0.9992	0.9992	0.999	0.9994	<b>0.9994</b>	1
24	0.9975	0.9975	0.9968	0.9977	0.9966	0.9959	0.9973	<b>0.9976</b>	2
Avg.	0.9988	0.9988	0.9985	0.9989	0.9981	0.998	0.9987	<b>0.999</b>	1
Diff.	0.0003	0.0002	0.0006	0.0001	0.0009	0.001	0.0003		

**Table 4** ZE results on the Kodak dataset for several demosaicking methods (the best performances are marked in bold).

Image	AFD	VCD	HOI	ESF	EDAEP	VDI	MLRI	BD	Rank
1	0.384	0.348	0.396	0.299	0.461	0.442	0.416	<b>0.276</b>	1
2	0.241	0.241	0.239	0.232	0.268	0.257	0.24	<b>0.214</b>	1
3	0.128	0.122	0.127	0.111	0.146	0.142	0.128	<b>0.106</b>	1
4	0.223	0.229	0.233	0.218	0.26	0.259	0.226	<b>0.209</b>	1
5	0.308	0.314	0.307	0.294	0.383	0.338	0.324	<b>0.278</b>	1
6	0.297	0.247	0.266	<b>0.206</b>	0.322	0.312	0.268	0.207	2
7	0.145	0.158	0.147	0.149	0.181	0.158	0.153	<b>0.134</b>	1
8	0.403	0.373	0.401	0.338	0.476	0.422	0.425	<b>0.326</b>	1
9	0.181	0.172	0.2	<b>0.16</b>	0.229	0.221	0.218	0.164	2
10	0.171	0.172	0.191	0.161	0.218	0.21	0.2	<b>0.16</b>	1
11	0.25	0.231	0.238	0.194	0.282	0.273	0.24	<b>0.193</b>	1
12	0.151	0.134	0.15	<b>0.114</b>	0.176	0.169	0.151	0.117	2
13	0.475	0.45	0.495	0.408	0.538	0.537	0.504	<b>0.397</b>	1
14	0.282	0.261	0.277	0.234	0.337	0.316	0.289	0.234	1
15	0.212	0.213	0.226	0.204	0.245	0.238	0.23	<b>0.203</b>	1

**Table 4** (Continued).

Image	AFD	VCD	HOI	ESF	EDAEP	VDI	MLRI	BD	Rank
16	0.209	0.157	0.181	<b>0.128</b>	0.223	0.213	0.178	0.151	2
17	0.174	0.171	0.189	0.156	0.227	0.215	0.193	<b>0.153</b>	1
18	0.346	0.352	0.367	0.341	0.4	0.392	0.371	<b>0.334</b>	1
19	0.273	0.254	0.285	0.233	0.335	0.304	0.29	<b>0.231</b>	1
20	0.243	0.246	0.253	0.232	0.274	0.259	0.248	<b>0.231</b>	1
21	0.285	0.273	0.299	<b>0.244</b>	0.335	0.318	0.308	0.246	2
22	0.298	0.316	0.315	0.299	0.339	0.32	0.307	<b>0.286</b>	1
23	0.154	0.166	0.161	0.16	0.173	0.165	0.161	<b>0.151</b>	1
24	0.259	0.261	0.27	0.241	0.319	0.284	0.281	<b>0.237</b>	1
Avg.	0.254	0.244	0.259	0.223	0.298	0.282	0.265	<b>0.218</b>	1
Diff.	0.036	0.026	0.041	0.005	0.08	0.064	0.046		



**Fig. 7** (a) Part of the original Kodak #8 image. Perceived image quality comparison using various deinterlacing methods: (b) AFD, (c) VCD, (d) HOI, (e) ESF, (f) EDAEP, (g) VDI, (h) MLRI, and (i) the proposed method.

In terms of the S-CIELAB  $\Delta E^*$  measure in the Kodak dataset, the BD method provides higher performance on average than all the other methods (AFD, VCD, HOI, ESF, EDAEP, VDI, and MLRI) with factors of 0.0522, 0.0450, 0.0703, 0.0337, 0.1789, 0.1319, and 0.0612, respectively, as shown in Table 2.

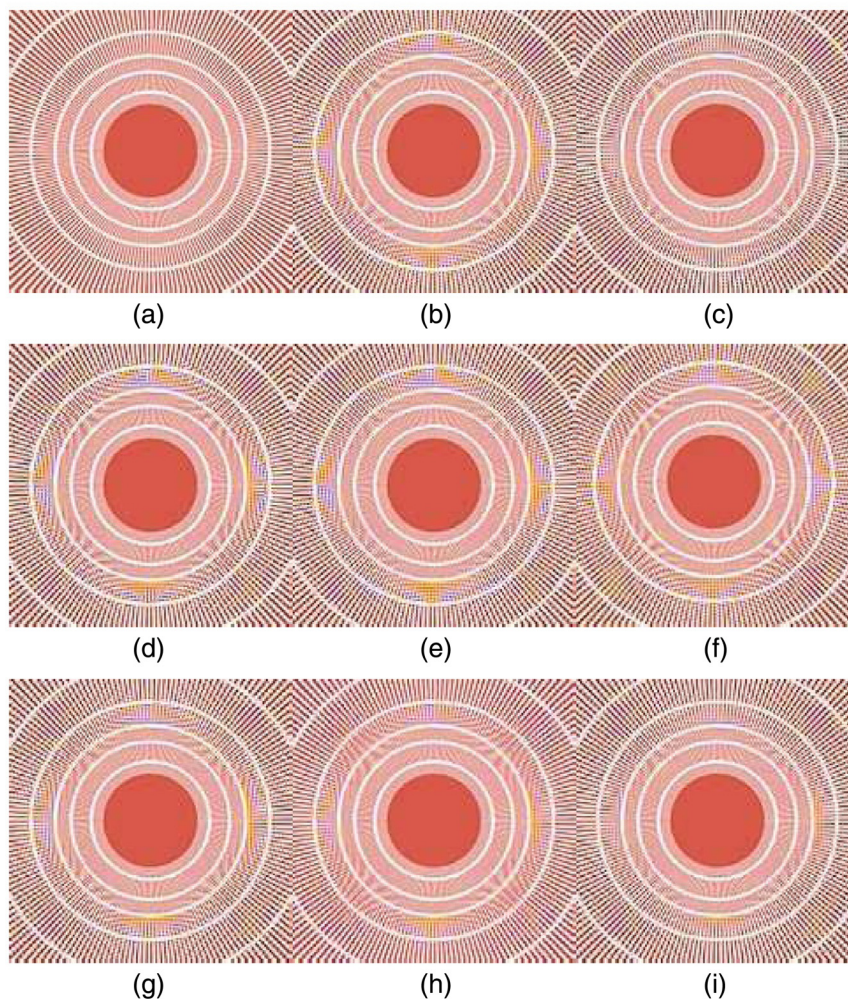
To further evaluate the performance of the BD method, we adopted a third metric called the FSIM index.<sup>18</sup> The best FSIM result is 1.0, whereas a smaller FSIM indicates poor visual quality. Table 3 presents the numerical FSIM results on the Kodak dataset for various benchmarks. The BD method outperforms the other methods (AFD, VCD, HOI, ESF, EDAEP, VDI, and MLRI) at 0.0003, 0.0002, 0.0006, 0.0001, 0.0009, 0.0010, and 0.0003 on average, respectively.

The ZE is one of the annoying artifacts in demosaicking. Lu and Tan introduced a measurement to evaluate the existence of the ZE. The result of the ZE represents the percentage of the pixel containing the ZE. Thus, the smaller the number is, the fewer the images containing the ZE. Table 4 shows that the BD method generated the smallest ZE. The BD method outperforms the other methods (AFD, VCD, HOI, ESF, EDAEP, VDI, and MLRI) at 0.036, 0.026, 0.041, 0.005, 0.080, 0.064, and 0.046 on average, respectively.

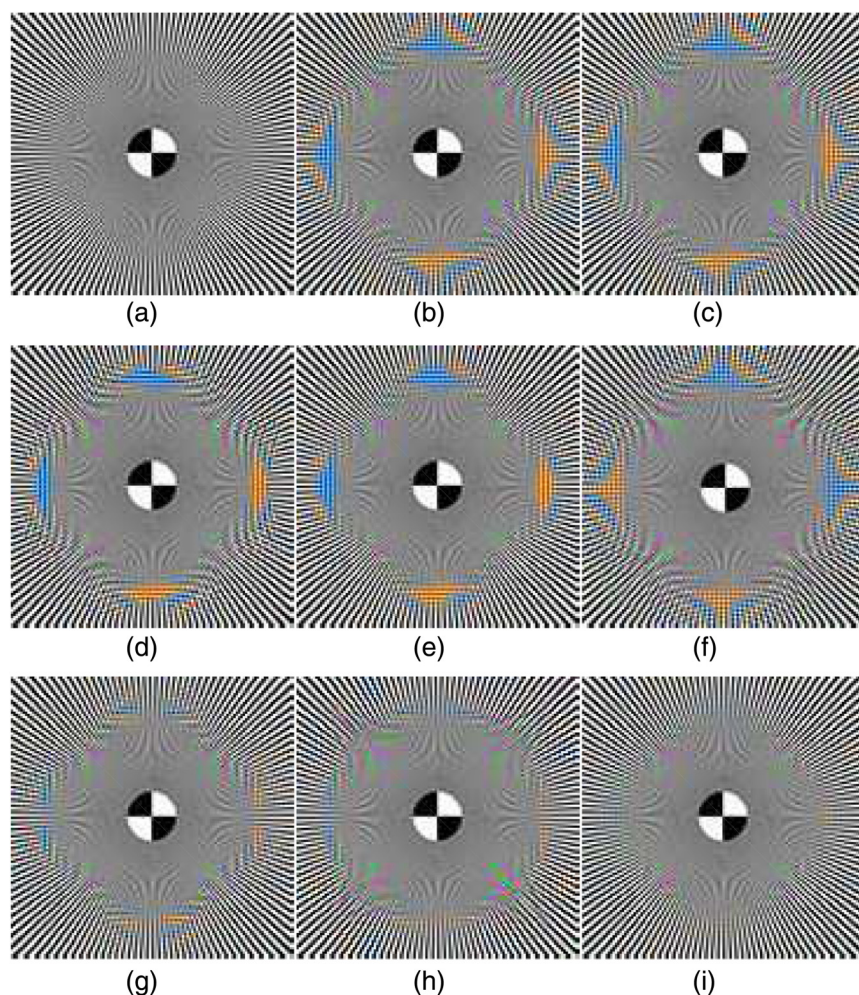
### 3.2 Visual Performance Analysis

In this section, we provide the subjective performance comparison between the BD method and the conventional methods. Figures 7–9 show the reconstructed images and their subjective performances in the visual comparison. Aliasing occurs near the image details because of signal information loss and poor restoration of high-frequency components. Therefore, the focus of the evaluation of the reconstruction of images should be the quality of the image edges. Figure 7 shows the zoomed demosaicking images on Kodak #8. From Fig. 7, we could see that our proposed algorithm has better visual performance at the white color reconstruction. One of evidences is that other conventional methods results show orange-blue color artifact. For example, the VDI method shows good visual quality seen in Fig. 7(g); however, compared with other subjective performances, VDI is much poorer than our algorithm, such as CPSNR, VDI is almost 4 dB less than our proposed algorithm.

To evaluate subjective performance in high-frequency images, we adopted a color radial resolution chart.<sup>16</sup> We tested eight methods on the TE216 and TE253 images, and the results are shown in Figs. 8 and 9. We noticed that the proposed method provides the best visual quality in



**Fig. 8** (a) Original TE216 image. Perceived image quality comparison using various deinterlacing methods: (b) AFD, (c) VCD, (d) HOI, (e) ESF, (f) EDAEP, (g) VDI, (h) MLRI, and (i) the proposed method.



**Fig. 9** (a) Original TE253 image. Perceived image quality comparison using various deinterlacing methods: (b) AFD, (c) VCD, (d) HOI, (e) ESF, (f) EDAEP, (g) VDI, (h) MLRI, and (i) the proposed method.

**Table 5** Subjective performance assessment by observers' evaluation.

Method	VCD	HOI	ESF	EDAEP	EDUSC	BD	Rank
Score	9.2098	9.2404	9.118	9.3016	9.3475	9.4546	1

high-frequency areas, whereas the conventional methods still showed orange-purple artifacts.

We have requested 20 observers to assess the quality of the result images that are produced by proposed method and the conventional methods. All observers provide a score for each result image, from 0 to 10. If the result image is identical to the original image, score 10 is provided. If result image is poorer than the original image, lower score is provided. Table 5 shows average scores evaluated by the 20 observers. From Table 5, we could conclude that our proposed method has highest favor from the observers' evaluation.

### 3.3 Performance Comparison in the Noise Condition

Noise is introduced during image acquisition from the sensor. To evaluate the robustness of the proposed method, we

conducted experiments on Kodak images contaminated by applying Gaussian and Poisson noises. Five selected conventional methods were used for comparison: VCD, HOI, ESF, EDAEP, and EDUSC.

#### 3.3.1 Gaussian noise case

Table 6 shows the CPSNR results under the condition of Gaussian noise. Our approach showed the best CPSNR, providing a gain of 0.195 dB compared with the ESF. The proposed method obtained 5 out of 24 images in the second-order in the noise-free images and two images (#16 and #19) in the second-order in the Gaussian noisy images.

#### 3.3.2 Poisson noise case

Some denoising approaches are based on the assumption that noise is Gaussian distributed. This assumption has been widely accepted, but recent research on denoising has considered noise as Poisson distributed given the improvement of the sensor with a reduced fixed pattern noise. The dominant noise type is the photon shot noise, which is strongly related to signal. Therefore, to verify the performance of the proposed algorithm in noisy images, we show the results for the Poisson noisy images of the Kodak dataset. Table 7 presents the CPSNR results on the demosaicking algorithms

**Table 6** CPSNR results on the Kodak dataset with Gaussian noise ( $\sigma = 10$ , the best performances are marked in bold).

	VCD	HOI	ESF	EDAEP	EDUSC	BD	Rank
1	32.861	32.379	33.247	31.694	32.709	<b>33.328</b>	1
2	33.379	33.629	33.457	33.443	33.485	<b>33.755</b>	1
3	33.792	33.986	33.794	33.897	33.898	<b>34.071</b>	1
4	33.465	33.654	33.409	33.562	33.564	<b>33.658</b>	1
5	32.692	32.817	32.636	32.164	32.848	<b>33.116</b>	1
6	33.297	33.186	33.634	32.572	32.913	<b>33.666</b>	1
7	33.672	33.981	33.699	33.793	33.859	<b>33.981</b>	1
8	32.17	31.639	32.583	30.584	31.654	<b>32.731</b>	1
9	33.801	33.927	33.836	33.737	33.842	<b>34.045</b>	1
10	33.754	33.912	33.791	33.744	33.847	<b>34.036</b>	1
11	33.237	33.231	33.444	32.754	33.225	<b>33.587</b>	1
12	33.906	34.05	33.976	33.875	33.897	<b>34.198</b>	1
13	31.556	30.566	32.142	29.767	31.291	<b>32.151</b>	1
14	32.435	32.663	32.114	32.34	32.635	<b>32.876</b>	1
15	33.471	33.477	33.375	33.468	33.509	<b>33.658</b>	1
16	33.823	33.863	<b>33.992</b>	33.542	33.592	33.705	4
17	33.626	33.674	33.701	33.469	33.711	<b>33.905</b>	1
18	32.517	32.434	32.74	31.993	32.682	<b>32.907</b>	1
19	33.426	33.381	<b>33.555</b>	32.863	33.319	33.542	2
20	34.371	34.431	34.459	34.105	34.484	<b>34.644</b>	1
21	33.126	32.927	33.408	32.443	33.116	<b>33.421</b>	1
22	32.828	32.992	32.927	32.733	32.985	<b>33.261</b>	1
23	33.856	34.118	33.787	34.037	33.973	<b>34.135</b>	1
24	31.707	31.391	31.839	31.169	31.639	<b>31.85</b>	1
Avg.	33.199	33.18	33.314	32.823	33.195	<b>33.509</b>	1
Diff.	0.311	0.33	0.195	0.687	0.315		

**Table 7** CPSNR results on the Kodak dataset with Poisson noise ( $\sigma = 10$ , the best performances are marked in bold).

	VCD	HOI	ESF	EDAEP	EDUSC	BD	Rank
1	37.946	36.255	39.257	34.703	37.37	<b>39.889</b>	1
2	39.905	40.282	40.249	39.304	40.04	<b>40.921</b>	1
3	41.447	41.64	41.481	40.877	41.637	<b>42.262</b>	1
4	40.044	40.047	39.789	39.459	40.098	<b>40.502</b>	1
5	37.441	37.333	37.238	35.582	37.674	<b>38.306</b>	1
6	39.202	38.223	40.226	36.453	37.502	<b>40.346</b>	1
7	41.041	41.569	41.076	40.438	41.61	<b>41.933</b>	1
8	35.94	34.591	36.775	32.679	34.784	<b>37.18</b>	1
9	41.388	41.104	41.471	39.988	41.221	<b>41.849</b>	1
10	41.184	41.104	41.303	40.102	41.282	<b>41.747</b>	1
11	39.297	38.641	40.006	37.154	38.921	<b>40.209</b>	1
12	41.512	41.35	41.803	40.359	41.055	<b>42.153</b>	1
13	34.569	32.631	35.814	31.388	33.961	<b>36.014</b>	1
14	36.68	36.845	35.857	36.019	37.052	<b>37.517</b>	1
15	39.134	38.587	38.751	38.356	38.899	<b>39.248</b>	1
16	42.265	41.384	<b>43.011</b>	39.742	40.302	42.51	2
17	40.807	40.259	41.17	39.194	40.922	<b>41.323</b>	1
18	36.853	36.255	37.45	35.219	37.133	<b>37.575</b>	1
19	39.662	39.145	40.478	37.326	39.335	<b>40.643</b>	1
20	40.159	39.785	40.455	38.637	40.338	<b>40.69</b>	1
21	38.541	37.47	39.521	36.127	38.317	<b>39.589</b>	1
22	37.593	37.582	37.905	36.812	37.885	<b>38.426</b>	1
23	41.737	42.174	41.332	41.335	41.866	<b>42.299</b>	1
24	34.861	33.973	<b>35.071</b>	33.559	34.546	34.93	2
Avg.	39.134	38.676	39.479	37.534	38.906	<b>39.919</b>	1
Diff.	0.786	1.243	0.441	2.385	1.013		

for Poisson noisy images. The proposed algorithm surpasses other conventional methods, and it improves images by 0.441 dB compared with the ESF. Similar to the Gaussian case, only 2 out of 24 images are ranked in the second-order for our proposed method, and the remaining ones are ranked in the first-order.

### 3.4 Performance Test in a High-Frequency Area

To compare visual performance in high-frequency images, we adopted two original high-frequency images: TE216 and TE253.<sup>16</sup> We tested seven methods on both images shown in Figs. 6(b) and 6(c). Figures 8 and 9 show the

reconstructed images of TE216 and TE253. The proposed method provides the best visual quality in the high-frequency area, followed by the MLRI method. It can be found from Fig. 8 that the VCD method shows the best visual quality in the high-frequency area, followed by the proposed method. For Fig. 9 image, the proposed method outperformed all the other benchmark methods. Although the proposed method provides the best objective performance, it is well known that the objective metrics (CPSNR, S-CIELAB, and FSIM) do not always rank quality of an image in the same way that observer does. There are many other factors considered by the human visual system and the brain, and therefore the perceived result could be different.

## 4 Conclusion

In this study, we proposed a demosaicking algorithm using an HBE in consideration of both pointwise and patchwise similarities. We introduced a Bayesian estimator to formulate the demosaicking model with MAP measurement. Our proposed method has superior performance in both noise-free and noisy images compared with conventional methods in both objective and subjective performances.

## Acknowledgments

This research was supported by Post-Doctor Research Program (2015) through the Incheon National University (INU), Incheon, South Korea.

## References

1. B. Leung, G. Jeon, and E. Dubois, "Least-squares luma-chroma demultiplexing algorithm for Bayer demosaicking," *IEEE Trans. Image Process.* **20**(7), 1885–1894 (2011).
2. G. Jeon and E. Dubois, "Demosaicking of noisy Bayer-sampled color images with least-squares luma-chroma demultiplexing and noise level estimation," *IEEE Trans. Image Process.* **22**(1), 146–156 (2013).
3. J. F. Hamilton and J. E. Adams, "Adaptive color plane interpolation in single sensor color electronic camera," U.S. Patent 5,629,734 (1997).
4. N. X. Lian et al., "Adaptive filtering for color filter array demosaicking," *IEEE Trans. Image Process.* **16**(10), 2515–2525 (2007).
5. K.-H. Chung and Y.-H. Chan, "Color demosaicing using variance of color differences," *IEEE Trans. Image Process.* **15**(10), 2944–2955 (2006).
6. J. S. J. Li and S. Randhawa, "Color filter array demosaicking using high-order interpolation techniques with a weighted median filter for sharp color edge preservation," *IEEE Trans. Image Process.* **18**(9), 1946–1957 (2009).
7. I. Pekkucuksen and Y. Altunbasak, "Edge strength filter based color filter array interpolation," *IEEE Trans. Image Process.* **21**(1), 393–397 (2012).
8. L. Chang and Y.-P. Tam, "Effective use of spatial and spectral correlations for color filter array demosaicing," *IEEE Trans. Consum. Electron.* **50**(1), 355–365 (2004).
9. W.-J. Chen and P.-Y. Chang, "Effective demosaicking algorithm based on edge property for color filter arrays," *Digital Signal Process.* **22**(1), 163–169 (2012).
10. X. Chen, G. Jeon, and J. Jeong, "Voting-based directional interpolation method and its application to still color image demosaicking," *IEEE Trans. Circuits Syst. Video Technol.* **24**(2), 255–262 (2014).
11. C.-Y. Su and W.-C. Kao, "Effective demosaicing using subband correlation," *IEEE Trans. Consum. Electron.* **55**(1), 199–204 (2009).
12. D. Kiku et al., "Minimized-Laplacian residual interpolation for color image demosaicking," *Proc. SPIE* **9023**, 90230L (2014).
13. F. Godtliebsen, E. Spjøtvoll, and J. S. Marron, "A nonlinear Gaussian filter applied to images with discontinuities," *J. Nonparametric Stat.* **8**, 21–43 (1997).
14. I. Pekkucuksen and Y. Altunbasak, "Multiscale gradients-based color filter array interpolation," *IEEE Trans. Image Process.* **22**(1), 157–165 (2013).
15. "Kodak lossless true color image suite," <http://r0k.us/graphics/kodak/> (25 April 2018).
16. "Image engineering," <http://www.image-engineering.de/> (25 April 2018).
17. X. Zhang and B. A. Wandell, "A spatial extension of CIELAB for digital color image reproduction," *J. Soc. Inf. Disp.* **5**(1), 61–63 (1997).
18. L. Zhang et al., "FSIM: a feature similarity index for image quality assessment," *IEEE Trans. Image Process.* **20**(8), 2378–2386 (2011).

**Jin Wang** received her BS degree in mathematics and applied mathematics from Zhejiang University, Hangzhou, China, and her MS and PhD degrees in electronic communications engineering from Hanyang University, Seoul, Korea, in 2007 and 2013, respectively. Currently, she is an associate professor at Xidian University in Xi'an, Shaanxi, China. She has been a researcher in the Department of Electronics and Computer Engineering, Hanyang University and the Department of Embedded Systems Engineering, Incheon University.

**Jiayi Wu** received his BS degree in electrical engineering from Xidian University, Xi'an, China, in 1996, his MS degree from National Time Service Center, the Chinese Academy of Sciences in 2002, and his PhD in electrical engineering from Xidian University in 2005. Currently, he is a professor at Xidian University, Xi'an, China. His current research interests include image processing, still image coding, hyperspectral/multispectral image compression, communication, and high performance computing.

**Zhensen Wu** received his BSc degree in applied physics from Xi'an Jiaotong University, Xi'an, China, in 1969, and his MSc degree in space physics from Wuhan University, Wuhan, China, in 1981. Currently, he is a professor at Xidian University, Xi'an, China. From 1995 to 2001, he was invited multiple times as a visiting professor to Rouen University, France, for implementing joint study of two projects supported by the Sino-France Program for Advanced Research.

**Marco Anisetti** is an assistant professor at the Università degli Studi di Milano. His research interests are in the area of computational intelligence and its application to the design of complex systems and services. He investigates/has investigated the adoption of advanced techniques based on Kalman filters, rough sets, and fuzzy set theories for: (i) low-level functionalities like display management, advanced image processing and mobile geolocation, and (ii) high-level applications in the areas of humanized computing, human machine interaction and ambient intelligence.

**Gwanggil Jeon** received his BS, MS, and PhD degrees from Hanyang University, in 2003, 2005, and 2008, respectively. From 2009 to 2011, he was a postdoctoral fellow at University of Ottawa, and from 2011 to 2012, he was an assistant professor at Niigata University. Currently, he is a professor at Incheon National University. His research interests fall under the umbrella of image processing, particularly image compression, motion estimation, demosaicking, and image enhancement as well as computational intelligence such as fuzzy and rough sets theories.