# Multiscale Gradients-Based Color Filter Array Interpolation

Ibrahim Pekkucuksen, *Member, IEEE*, and Yucel Altunbasak, *Fellow, IEEE*

***Abstract*— Single sensor digital cameras use color filter arrays to capture a subset of the color data at each pixel coordinate. Demosaicing or color filter array (CFA) interpolation is the process of estimating the missing color samples to reconstruct a full color image. In this paper, we propose a demosaicing method that uses multiscale color gradients to adaptively combine color difference estimates from different directions. The proposed solution does not require any thresholds since it does not make any hard decisions, and it is noniterative. Although most suitable for the Bayer CFA pattern, the method can be extended to other mosaic patterns. To demonstrate this, we describe its application to the Lukac CFA pattern. Experimental results show that it outperforms other available demosaicing methods by a clear margin in terms of CPSNR and S-CIELAB measures for both mosaic patterns.**

***Index Terms*— Color filter array (CFA) interpolation, demosaicing, directional interpolation, multiscale color gradients.**

## I. Introduction

MOST digital cameras use color filter arrays instead of beam splitters and this design choice leads to the capture of only a subset of the image data. The color channels filtered out by the CFA pattern layout need to be estimated using the recorded channel values. The process of interpolating a full color image from a mosaicked single channel input is called demosaicing or CFA interpolation. The CFA pattern layout plays an important role in the design of a CFA interpolation algorithm. Many different CFA patterns have been proposed over the years. While some of these are comprised of pure RGB channels, like Bayer and Lukac patterns shown in Fig. 1 [1], [2], others feature linear combinations of RGB channels [3], [4].

Demosaicing is an important part of the image processing pipeline in digital cameras. The failure of the employed demosaicing algorithm can degrade the overall image quality considerably. That is why it has been an active research area for many years. Although there has been recent efforts to introduce generalized demosaicing algorithms, most demosaicing solutions in the literature are developed for the Bayer pattern.

The simplest way to address the demosaicing problem would be to treat each color channel seperately and interpolate missing samples using a spatially invariant method such as bilinear or bicubic interpolation. However, such a solution would lead to false color artifacts wherever there is a sudden color change. The quality can be improved by applying the interpolation over color differences to take advantage of the correlation between the color channels. However, the lack of spatial adaptiveness would still limit the interpolation performance.

A sophisticated demosaicing solution needs to be able to use both spatial and spectral correlations effectively. The spectral correlation can be put to use by the constant color ratio rule [5], [6] or the constant color difference rule [7], [8]. It is assumed that these rules would be valid inside a given object over a local area. However, a natural image consists of many objects with boundaries between them. Too much dependence on spectral correlation could cause zipper artifacts across these object boundaries.

An early demosaicing method proposed in [7] used derivatives of chrominance samples in initial green channel interpolation, and this idea is borrowed by many subsequent algorithms. Various demosaicing algorithms proposed directional interpolation with different decision rules [7]–[10]. For instance, Chung et al. [11] used variance of color differences to make a hard direction decision. On the other hand, the authors of [12] proposed a soft direction decision based on the Linear Minimum Mean Square Error Estimation (LMMSE) framework. Here, the directional color differences are considered as noisy observations of the actual color difference and they are combined optimally. Paliy et al. [13] improved this directional approach with scale adaptive filtering based on local polynomial approximation (LPA). Another interesting directional approach is to perform interpolation in both directions and then make a posteriori decision. Hirakawa et al. [14] used local homogeneity of the directional interpolation results and Menon et al. [15] used color gradients over a local window as the decision criteria.

The demosaicing problem has been studied in the frequency domain as well. Gloztbach et al. proposed using high frequency components extracted from green channel to improve red and blue channel components that are more susceptible to aliasing [16]. In [17], Gunturk et al. proposed an alternating projections scheme using the strong inter-channel correlation in high frequency subbands. Since the method is iterative, it required a high number of calculations. Another method [18] proposed filtering the input mosaicked color components together to preserve the high frequency components better. Here, the luminance is estimated by a fixed 5 by 5 filter at green pixel locations and by an adaptive filter at red and blue channel locations. The full range of demosaicing

Manuscript received August 8, 2011; revised April 8, 2012; accepted June 1, 2012. Date of publication July 30, 2012; date of current version December 20, 2012. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Wan-Chi Siu.

The authors are with the Department of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA 30332 USA (e-mail: ibrahimp@gatech.edu; yucel@ece.gatech.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TIP.2012.2210726

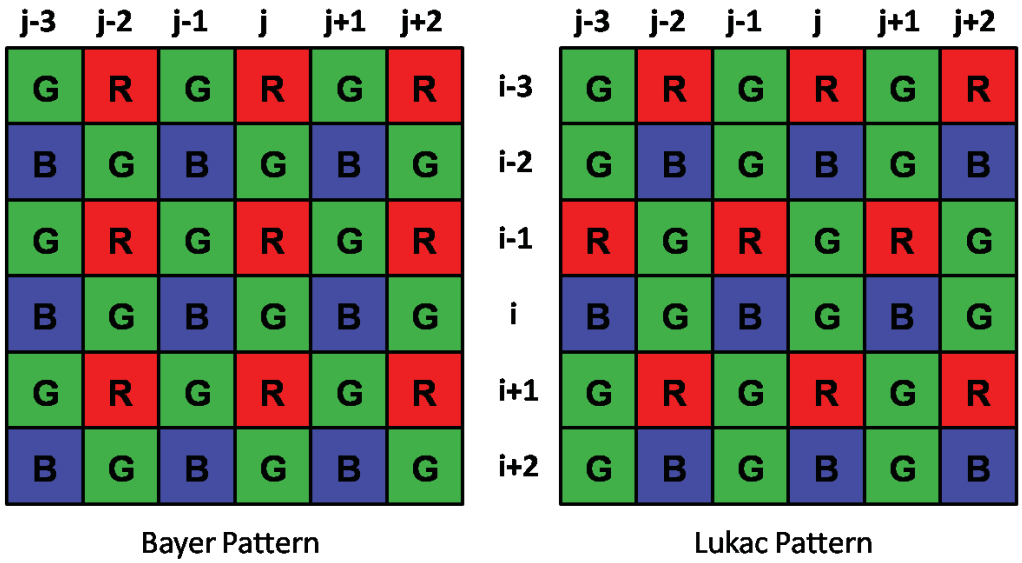


Fig. 1. Bayer and Lukac mosaic patterns.

methods in the literature is too diverse to cover in this paper. A comprehensive overview with objective comparison results can be found in a recent survey paper [19] by Li et. al. and the PhD thesis of Menon [21]. An overview of linear approaches to the problem can be found in [20].

The rest of the paper is organized as follows. Section II first gives the background of the proposed algorithm and then describes it in detail. Section III presents experimental results, and Section IV concludes the paper.

## II. Proposed Algorithm

### A. Algorithm Background

Gradients are useful for extracting directional data from digital images. Several demosaicing methods including a recent integrated gradients method proposed in [22] made use of them. We demonstrated in [23] that the gradients of color difference signals could be valuable features to adaptively combine directional color difference estimates. In this method, the horizontal and vertical color difference estimates are blended based on the ratio of the total absolute values of vertical and horizontal color difference gradients over a local window.

The first step of the algorithm is to get initial directional color channel estimates. The Bayer pattern is comprised of blue&green and red&green rows and columns as depicted in Fig. 1. For red&green rows and columns in the input mosaic image, the directional estimates for the missing red and green pixel values are:

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

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

where $H$ and $V$ denote horizontal and vertical directions and $(i, j)$ is the pixel location. For every pixel coordinate, we now have a true color channel value and two directional estimates. By taking their difference, we get the directional color difference estimate:

$$\tilde{\Delta}^H_{g,r}(i,j) = \begin{cases} \tilde{G}^H(i,j) - R(i,j), & \text{if G is interpolated} \\ G(i,j) - \tilde{R}^H(i,j), & \text{if R is interpolated} \end{cases}$$
$$\tilde{\Delta}^V_{g,r}(i,j) = \begin{cases} \tilde{G}^V(i,j) - R(i,j), & \text{if G is interpolated} \\ G(i,j) - \tilde{R}^V(i,j), & \text{if R is interpolated} \end{cases} \quad (2)$$

where $\tilde{\Delta}^H_{g,r}$ stands for the horizontal difference estimate between green and red channels. The equations are similar for blue&green rows and columns. The generated horizontal and vertical color difference maps are shown in Figure 2. As mentioned above, the directional estimates are combined adaptively using the color difference gradients. The absolute color difference gradients at pixel coordinates (i, j) are given by:

$$D^H(i,j) = |\tilde{\Delta}^H(i,j-1) - \tilde{\Delta}^H(i,j+1)|$$
$$D^V(i,j) = |\tilde{\Delta}^V(i-1,j) - \tilde{\Delta}^V(i+1,j)|. \quad (3)$$

It could be argued that the performance of such an algorithm relies on its ability to successfully combine directional estimates. The color difference gradients calculated above are used to find weights for each direction. The horizontal color difference gradient equation above can be written in terms of

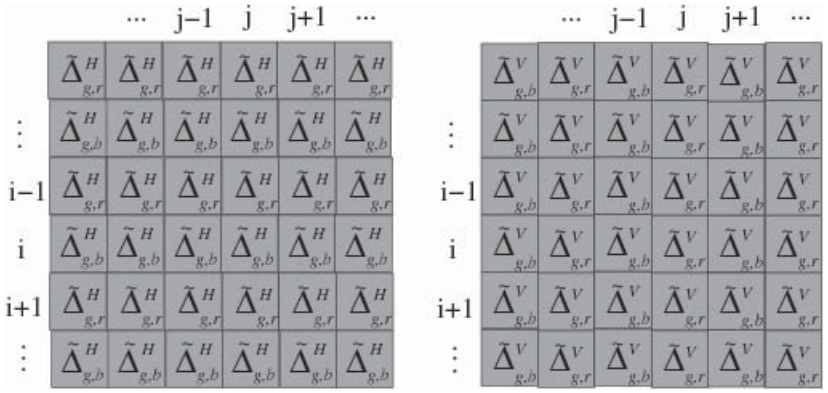


Fig. 2. Horizontal and vertical color difference maps.

red and green pixel values as follows:

$$D^H(i,j) = \left|(G(i,j-1) - \tilde{R}^H(i,j-1)) - (G(i,j+1) - \tilde{R}^H(i,j+1))\right| = \left|\left(\frac{2.G(i,j-1)+G(i,j-3)+G(i,j+1)}{4} - \frac{R(i,j-2)+R(i,j)}{2}\right) - \left(\frac{2.G(i,j+1)+G(i,j-1)+G(i,j+3)}{4} - \frac{R(i,j)+R(i,j+2)}{2}\right)\right|. \quad (4)$$

We observe that there are $R(i,j)$ terms present and they cancel out each other. Rearranging with respect to different color channels leaves us with:

$$D^H(i,j) = \left|\frac{R(i,j+2)-R(i,j-2)}{2} - \frac{(G(i,j+3)+G(i,j+1))-(G(i,j-3)+G(i,j-1))}{4}\right|. \quad (5)$$

There are two important observations that we made on the equation above. First, our color difference gradient corresponds to taking the difference between the available color channel values two pixels away from the target pixel, doing the same operation in terms of the other color channel by using simple averaging, and then finding the difference between these two operations as illustrated in the top portion of Figure 3. If these two color channels are changing in parallel with each other along this direction, then the resulting absolute value would be small. On the other hand, if there is an abrupt color change, then the result would be large and the color difference estimate along this direction would be given a small weight in combined color difference calculation. Our second and more important observation is that, we can do these same operations at half the scale:

$$D^h(i,j) = \left|\frac{G(i,j+1)-G(i,j-1)}{2} - \frac{(R(i,j+2)+R(i,j))-(R(i,j-2)+R(i,j))}{4}\right|, \quad (6)$$

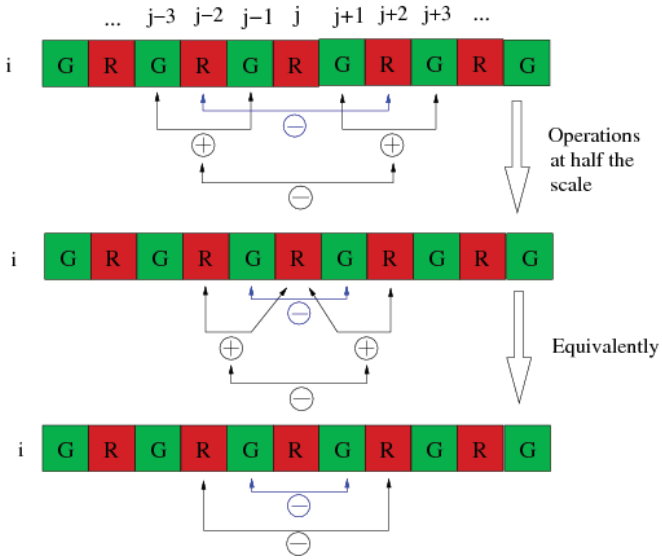


Fig. 3. Relationship between the color difference gradients equation and the multiscale gradients equation.

where $D^h(i,j)$ denotes the horizontal estimation at half the scale. A smaller scale is more desirable because it allows the local color dynamics to be captured at a better resolution. Note that the available color channels are replaced at this scale, but we are still performing the same operations: We take the difference between the available color channel values one pixel (instead of two pixels) away from the target pixel, we do the same operation in terms of the other channel by using its closest samples, and then we take the difference between these two. At this scale, the $R(i,j)$ terms cancel each other out and we are left with:

$$D^h(i,j) = \left|\frac{G(i,j+1)-G(i,j-1)}{2} - \frac{R(i,j+2)-R(i,j-2)}{4}\right|. \quad (7)$$

We observe that the first part of this equation is the green channel gradient, and the second part is the red channel gradient at twice the scale normalized by the distance between their operands as shown in the bottom part of Figure 3.

Fig. 4. 24 image Kodak test set.

Like the color difference gradient equation (Equation no. 5), this equation checks whether different color channel pixels along this direction are changing in agreement with each other or not. However, we expected this new equation to be more successful with combining the directional estimates because we capture the color dynamics at a more local level and we do it without resorting to any simple averaging.

The fact that this equation combines two different scales of gradients together gave us the idea that it should be possible to incorporate even more scales into the equation. However, since the locality will get weaker with each additional scale, the larger scales should contribute less to the result. The easiest way of doing that is to optimize the normalizing terms in the denominators. The final multiscale gradients equations for red&green rows and columns can be given as follows:

$$\begin{aligned} D^h(i,j) = \Bigg| & \frac{G(i,j+1)-G(i,j-1)}{2} \\ & - \frac{R(i,j+2)-R(i,j-2)}{N_1} \\ & + \frac{G(i,j+3)-G(i,j-3)}{N_2} \\ & - \frac{R(i,j+4)-R(i,j-4)}{N_3} + \cdots \Bigg| \\ D^v(i,j) = \Bigg| & \frac{G(i+1,j)-G(i-1,j)}{2} \\ & - \frac{R(i+2,j)-R(i-2,j)}{N_1} \\ & + \frac{G(i+3,j)-G(i-3,j)}{N_2} \\ & - \frac{R(i+4,j)-R(i-4,j)}{N_3} + \cdots \Bigg|, \end{aligned} \quad (8)$$

where the $N_i$ terms are the normalizers. The equations are similar for blue&green rows and columns.

### B. Initial Green Channel Interpolation

Like most demosaicing methods designed for the Bayer pattern, our algorithm starts with interpolating the green channel. After updating the initial green channel interpolation results in one pass, the red and blue channels are filled in using the constant color difference assumption. The ratio between the vertical and horizontal multiscale gradients results over a local window is employed at every stage.

For initial green channel interpolation, we have directional color difference estimates around every green pixel to be interpolated as given in Equation (2) and we combine them adaptively:

$$\begin{aligned} \hat{\Delta}_{g,r}(i,j) = & [w_V.\mathbf{f}.\tilde{\Delta}^V_{g,r}(i-1:i+1,j) \\ & + w_H.\tilde{\Delta}^H_{g,r}(i,j-1:j+1).\mathbf{f}']/w_C \\ w_C = & \; w_V + w_H \\ \mathbf{f} = & \; [1/4 \;\; 2/4 \;\; 1/4]. \end{aligned} \quad (9)$$

For a local window size of 5 by 5, the weight for each direction is calculated as follows:

$$\begin{aligned} w_V &= 1/\left(\sum_{k=i-2}^{i+2}\sum_{l=j-2}^{j+2} D^v(k,l)\right)^2 \\ w_H &= 1/\left(\sum_{k=i-2}^{i+2}\sum_{l=j-2}^{j+2} D^h(k,l)\right)^2. \end{aligned} \quad (10)$$

The division operation can be avoided by defining the weights as the denominators and exchanging them (The ratio of $1/a$ to $1/b$ is equal to the ratio of $b$ to $a$ provided that both are nonzero).

### C. Green Channel Update

After the directional color difference estimates are combined as explained in the previous section, we can directly calculate the green channel and move onto completing the other channels. However, it is possible to improve the green channel results by updating the initial color difference estimates. We consider the closest four neighbors to the target pixel with each one having its own weight:

$$\begin{aligned} \tilde{\Delta}_{g,r}(i,j) = & \; \hat{\Delta}_{g,r}(i,j).(1-w) \\ & + \big[w_N.\hat{\Delta}_{g,r}(i-2,j) + w_S.\hat{\Delta}_{g,r}(i+2,j) \\ & \quad + w_E.\hat{\Delta}_{g,r}(i,j-2) + w_W.\hat{\Delta}_{g,r}(i,j+2)\big] \\ & .w/w_T \\ w_T = & \; w_N + w_S + w_E + w_W. \end{aligned} \quad (11)$$

Again, the weights $(w_N, w_S, w_E, w_W)$ are calculated by finding the total multiscale color gradients over a local window. For a 3 by 5 window for horizontal and a 5 by 3 window for vertical components, the weight calculations can be given

as follows:

$$w_N = 1/\left(\sum_{k=i-4}^{i}\sum_{l=j-1}^{j+1} D^v(k,l)\right)^2$$
$$w_S = 1/\left(\sum_{k=i}^{i+4}\sum_{l=j-1}^{j+1} D^v(k,l)\right)^2$$
$$w_W = 1/\left(\sum_{k=i-1}^{i+1}\sum_{l=j-4}^{j} D^h(k,l)\right)^2$$
$$w_E = 1/\left(\sum_{k=i-1}^{i+1}\sum_{l=j}^{j+4} D^h(k,l)\right)^2. \quad (12)$$

Once the color difference estimate is finalized, we add it to the available target pixel to obtain the estimated green channel value:

$$\tilde{G}(i,j) = R(i,j) + \tilde{\Delta}_{g,r}(i,j)$$
$$\tilde{G}(i,j) = B(i,j) + \tilde{\Delta}_{g,b}(i,j). \quad (13)$$

### D. Red and Blue Channel Interpolation

For red and blue channel interpolation, we first complete the missing diagonal samples i.e. red pixel values at blue locations and blue pixel values at red locations. These pixels are interpolated using the 7 by 7 filter proposed in [13]:

$$p_{rb} = \begin{bmatrix} 0 & 0 & -1 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & 10 & 0 & 10 & 0 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & 10 & 0 & 10 & 0 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & -1 & 0 & 0 \end{bmatrix} \cdot \frac{1}{32}$$
$$\tilde{R}_{i,j} = \tilde{G}_{i,j} - \tilde{\Delta}_{g,r}(i-3:i+3, j-3:j+3) \otimes p_{rb}$$
$$\tilde{B}_{i,j} = \tilde{G}_{i,j} - \tilde{\Delta}_{g,b}(i-3:i+3, j-3:j+3) \otimes p_{rb}, \quad (14)$$

where $\otimes$ denotes element-wise matrix multiplication and subsequent summation.

The red and blue pixels at green locations are interpolated adaptively. In order to avoid repetitive weight calculations, we reuse the directional weights $(w_H, w_V)$ defined in Equation (10). The immediate vertical neighbors of a green pixel are either red or blue pixels. For the red pixel case the interpolation is carried out as follows:

$$\tilde{R}(i,j) = G(i,j) - \frac{w_V.(\tilde{G}(i-1,j) - R(i-1,j) + \tilde{G}(i+1,j) - R(i+1,j))}{2.(w_V + w_H)} - \frac{w_H.(\tilde{G}(i,j-1) - \tilde{R}(i,j-1) + \tilde{G}(i,j+1) - \tilde{R}(i,j+1))}{2.(w_V + w_H)}$$
$$\tilde{B}(i,j) = G(i,j) - \frac{w_V.(\tilde{G}(i-1,j) - \tilde{B}(i-1,j) + \tilde{G}(i+1,j) - \tilde{B}(i+1,j))}{2.(w_V + w_H)} - \frac{w_H.(\tilde{G}(i,j-1) - B(i,j-1) + \tilde{G}(i,j+1) - B(i,j+1))}{2.(w_V + w_H)}. \quad (15)$$

The equations for the blue vertical neighbor case are similar. With the completion of red and blue pixel values at green coordinates, we obtain the full color image.

### E. Application to the Lukac Pattern

Although designed for the Bayer mosaic pattern, the proposed method can be modified to be applied to other mosaic patterns. However, such an application may not be feasible for all mosaic patterns because of the restrictions dictated by the directional nature of our approach. When the modification is feasible, an important question would be whether the changes needed to comply with the new pattern layout lead to a significant performance loss or not. To find out if we can outperform other available solutions on a different pattern layout, we modified the proposed algorithm for the Lukac pattern.

Lukac mosaic pattern is similar to Bayer pattern in the sense that it consists of pure RGB components. When we shift every other row in a Bayer pattern by one pixel to either side, we obtain the Lukac pattern. Hence, the horizontal relationship between the pixels is still the same, but the vertical arrangement is significantly altered. As a result of this, it is not possible to take immediate vertical gradients anymore. However, we observe that we can take vertical gradients when we double the scale. So we modified our vertical multiscale

$$\tilde{R}^V(i,j) = \frac{\frac{R(i-1,j-1)+R(i-1,j+1)}{2} + R(i+1,j)}{2} + \frac{2.G(i,j) - \frac{G(i-2,j-1)+G(i-2,j+1)}{2} - \frac{G(i+2,j-1)+G(i+2,j+1)}{2}}{4}$$
$$\tilde{G}^V(i-1,j-1) = \frac{G(i-2,j-1) + \frac{G(i,j-2)+G(i,j)}{2}}{2} + \frac{2.R(i-1,j-1) - \frac{R(i-3,j-2)+R(i-3,j)}{2} - \frac{R(i+1,j-2)+R(i+1,j)}{2}}{4}. \quad (17)$$

gradients equation accordingly:

$$D^v(i,j) = \Bigg| \frac{G(i+2,j)-G(i-2,j)}{M_0} - \frac{R(i+4,j)-R(i-4,j)}{M_1} + \frac{G(i+6,j)-G(i-6,j)}{M_2} - \frac{R(i+8,j)-R(i-8,j)}{M_3} + \cdots \Bigg|, \quad (16)$$

where the $M_i$ terms are the normalizers.

The layout of the Lukac pattern also necessitates a change in vertical color difference estimation. Since all the required channel values are not available in the same column, we estimate the missing values by taking simple average using samples from adjacent columns, shown at the bottom of the previous page.

Another problem we faced with the Lukac pattern was the mismatch between vertical and horizontal color difference estimates at green channel coordinates. Namely, the calculated vertical and horizontal color differences at these locations belong to different color pairs. That is why we bring the needed vertical color difference estimate from the closest available resource:

$$\tilde{\Delta}^V_{g,b}(i,j) = G(i-1,j) - \tilde{B}^V(i-1,j)$$
$$\tilde{\Delta}^V_{g,r}(i-1,j) = G(i,j) - \tilde{R}^V(i,j).$$

Also, the combined color difference estimate equations are modified to bring the neighboring vertical estimates from two pixels away instead of one:

$$\begin{aligned} \hat{\Delta}_{g,r}(i-1,j-1) &= \big[w_V.\mathbf{f}_v.\tilde{\Delta}^V_{g,r}(i-3:i+1,j) \\ &\quad + w_H.\tilde{\Delta}^H_{g,r}(i-1,j-2:j).\mathbf{f}'_h\big]/w_C \\ w_C &= w_V + w_H \\ \mathbf{f}_h &= [1/4\ 2/4\ 1/4] \\ \mathbf{f}_v &= [1/4\ 0\ 2/4\ 0\ 1/4]. \end{aligned} \quad (18)$$

Similarly, we modify the green channel update equation as follows:

$$\begin{aligned} \tilde{\Delta}_{g,r}(i,j) &= \hat{\Delta}_{g,r}(i,j).(1-w) \\ &\quad +[w_N.(\hat{\Delta}_{g,r}(i-2,j-1) + \hat{\Delta}_{g,r}(i-2,j+1) \\ &\quad + \hat{\Delta}_{g,r}(i-4,j))/3 w_S.(\hat{\Delta}_{g,r}(i+2,j-1) \\ &\quad + \hat{\Delta}_{g,r}(i+2,j+1) + \hat{\Delta}_{g,r}(i+4,j))/3 \\ &\quad w_E.\hat{\Delta}_{g,r}(i,j-2) + w_W.\hat{\Delta}_{g,r}(i,j+2)].w/w_T \\ w_T &= w_N + w_S + w_E + w_W. \end{aligned} \quad (19)$$

And finally, the red and blue channel interpolation requires modification as well. We estimate the missing red and blue samples using the closest color difference estimates. For the red channel interpolation, the pixels on green&blue rows use estimates from three neighbors and the ones on green&red rows use four:

$$\begin{aligned} \tilde{R}(i,j) &= G(i,j) - \frac{\tilde{G}(i+1,j) - R(i+1,j)}{2} \\ &\quad - \frac{\tilde{G}(i-1,j-1) - R(i-1,j-1) + \tilde{G}(i-1,j+1) - R(i-1,j+1)}{4} \\ \tilde{R}(i,j-1) &= G(i,j-1) - \frac{\tilde{G}(i-1,j-1) - R(i-1,j-1)}{2} \\ &\quad - \frac{\tilde{G}(i+1,j-2) - R(i+1,j-2) + \tilde{G}(i+1,j) - R(i+1,j)}{4} \\ \tilde{R}(i-1,j) &= G(i-1,j) \\ &\quad - \frac{\tilde{G}(i-1,j-1) - R(i-1,j-1) + \tilde{G}(i-1,j+1) - R(i-1,j+1)}{2.5} \\ &\quad - \frac{\tilde{G}(i-3,j) - R(i-3,j) + \tilde{G}(i+1,j) - R(i+1,j)}{10}. \end{aligned} \quad (20)$$

Although we needed to make several changes to apply the algorithm to the Lukac pattern, the main structure of the method is maintained.

## III. Experimental Results

The performance of the proposed method is evaluated on the Kodak test set which consists of 24 images. Thumbnails of the test images are listed in Figure 4. The full color images are first downsampled using Bayer and Lukac patterns. Then, they were reconstructed using the proposed method for each pattern. The difference between the original and the reconstructed images is reported in terms of CPSNR and S-CIELAB error measures. Coordinates within 10 pixels from the border are excluded from the calculations. For the Bayer pattern, the results of the proposed method are compared to Alternating Projections (AP) [17], Variance of Color Differences (VCD) [11], Directional Linear Minimum Mean Square-Error Estimation (DLMMSE) [12], Local Polynomial Approximation (LPA) [13], Regularization Approaches to Demosaicing (RAD) [24], Gradient Based Threshold Free demosaicing (GBTF) [23], Least-Squares Luma-Chroma Demultiplexing (LSLCD) [25], and Integrated Gradients (IGD) [22] methods. LPA, DLMMSE, and VCD are the highest performing methods featured in a recent survey paper [19] on demosaicing. RAD, GBTF, LSLCD, and IGD are newer methods that are introduced after this survey paper. The CPSNR comparison results for the Bayer pattern are summarized in Table I. The proposed algorithm outperforms other methods on 22 images out of 24. Its average CPSNR is better than the closest method (IGD) by 0.37 dB. Similarly, it outperforms other methods in terms of S-CIELAB measure on all images except two. The S-CIELAB comparison results are given in Table II. The fence region from the lighthouse image is shown in Figure 5 for subjective quality comparison.

We calculated the computational complexity of the proposed algorithm in terms of the number of operations needed for every 2 by 2 GRBG block. Since our algorithm does not have conditional branches, its computational complexity is not image dependent. The algorithm requires 348 addition, 100 multiplication, 44 shift, 22 division, and 8 absolute value operations per 4 pixels. Based on these numbers, the complexity of

TABLE I
COMPARISON OF CPSNR ERROR MEASURE FOR DIFFERENT DEMOSAICING METHODS ON THE BAYER PATTERN

| No. | AP | VCD | DL | LPA | RAD | GBTF | IGD | LSLCD | Prop |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 37.80 | 38.61 | 38.52 | **40.46** | 38.29 | 39.62 | 40.09 | 38.68 | 39.87 |
| 2 | 39.65 | 40.42 | 40.93 | 41.33 | 39.85 | 41.53 | 41.06 | 40.89 | **41.77** |
| 3 | 41.63 | 42.71 | 42.75 | 43.47 | 42.12 | 43.30 | 43.42 | 42.48 | **43.72** |
| 4 | 40.16 | 40.62 | 40.57 | 40.86 | 40.69 | 40.70 | 40.68 | 40.96 | **41.13** |
| 5 | 37.58 | 37.97 | 38.10 | 37.54 | 38.02 | 38.42 | 38.43 | 38.42 | **39.05** |
| 6 | 38.64 | 40.09 | 40.27 | 40.93 | 39.85 | 41.05 | 41.15 | 40.42 | **41.38** |
| 7 | 41.84 | 42.26 | 42.39 | 43.02 | 42.44 | 42.90 | 42.70 | 42.97 | **43.51** |
| 8 | 35.16 | 36.40 | 36.08 | 37.13 | 36.03 | 37.02 | 37.46 | 35.70 | **37.56** |
| 9 | 41.89 | 43.14 | 43.14 | 43.49 | 42.40 | 43.62 | 43.56 | 42.81 | **43.96** |
| 10 | 42.00 | 42.48 | 42.53 | 42.67 | 42.61 | 42.93 | 42.82 | 42.55 | **43.20** |
| 11 | 39.26 | 39.96 | 40.09 | 40.53 | 39.71 | 40.89 | 40.81 | 40.34 | **41.36** |
| 12 | 42.64 | 43.57 | 43.53 | 43.98 | 43.20 | 44.15 | 44.25 | 43.49 | **44.45** |
| 13 | 34.38 | 34.92 | 34.81 | 36.09 | 34.85 | 35.49 | **36.14** | 35.06 | 36.00 |
| 14 | 35.84 | 37.06 | 37.03 | 36.97 | 36.12 | 37.22 | 37.33 | 37.10 | **37.97** |
| 15 | 39.43 | 39.84 | 39.87 | 40.09 | 39.89 | 40.02 | 39.92 | 40.12 | **40.30** |
| 16 | 41.89 | 43.75 | 43.83 | 43.99 | 43.26 | 44.48 | 44.61 | 44.09 | **44.86** |
| 17 | 41.32 | 41.49 | 41.79 | 41.80 | 41.67 | 42.09 | 41.97 | 41.77 | **42.32** |
| 18 | 37.19 | 37.18 | 37.52 | 37.42 | 37.51 | 37.84 | 37.75 | 37.66 | **38.22** |
| 19 | 39.53 | 41.03 | 41.04 | 41.51 | 40.19 | 41.82 | 41.84 | 40.78 | **42.17** |
| 20 | 40.72 | 41.17 | 41.27 | 41.44 | 41.00 | 41.83 | 41.76 | 41.29 | **42.16** |
| 21 | 38.79 | 39.23 | 39.17 | 39.63 | 39.24 | 39.77 | 40.14 | 39.28 | **40.31** |
| 22 | 37.72 | 38.21 | 38.46 | 38.49 | 38.30 | 38.85 | 38.63 | 38.73 | **39.05** |
| 23 | 42.02 | 43.05 | 43.30 | 43.89 | 42.36 | 43.65 | 43.33 | 43.28 | **44.02** |
| 24 | 34.74 | 35.15 | 35.52 | 35.37 | 35.56 | 35.62 | 35.36 | 35.62 | **35.69** |
| Avg | 39.24 | 40.01 | 40.10 | 40.50 | 39.80 | 40.62 | 40.63 | 40.19 | **41.00** |

TABLE II
COMPARISON OF S-CIELAB ERROR MEASURE FOR DIFFERENT DEMOSAICING METHODS ON THE BAYER PATTERN

| No. | AP | VCD | DL | LPA | RAD | GBTF | IGD | LSLCD | Prop |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 1.618 | 1.442 | 1.429 | **1.226** | 1.562 | 1.289 | 1.251 | 1.459 | **1.274** |
| 2 | 1.647 | 1.486 | 1.413 | 1.370 | 1.667 | 1.352 | 1.389 | 1.487 | **1.323** |
| 3 | 0.929 | 0.852 | 0.850 | 0.803 | 0.921 | 0.811 | 0.810 | 0.874 | **0.782** |
| 4 | 1.263 | 1.194 | 1.161 | 1.151 | 1.153 | 1.138 | 1.152 | 1.159 | **1.096** |
| 5 | 2.125 | 1.995 | 1.951 | 2.020 | 2.047 | 1.851 | 1.846 | 1.937 | **1.729** |
| 6 | 1.221 | 1.053 | 1.027 | 0.977 | 1.122 | 0.957 | 0.950 | 1.029 | **0.931** |
| 7 | 1.101 | 1.025 | 1.017 | 0.946 | 1.048 | 0.962 | 0.981 | 0.966 | **0.907** |
| 8 | 1.850 | 1.579 | 1.584 | 1.454 | 1.723 | 1.457 | 1.416 | 1.679 | **1.393** |
| 9 | 0.829 | 0.736 | 0.732 | 0.719 | 0.782 | 0.705 | 0.720 | 0.779 | **0.689** |
| 10 | 0.822 | 0.770 | 0.759 | 0.756 | 0.780 | 0.737 | 0.749 | 0.793 | **0.724** |
| 11 | 1.442 | 1.298 | 1.271 | 1.208 | 1.442 | 1.174 | 1.172 | 1.297 | **1.132** |
| 12 | 0.666 | 0.606 | 0.598 | 0.579 | 0.644 | 0.569 | 0.571 | 0.615 | **0.558** |
| 13 | 2.547 | 2.427 | 2.396 | 2.263 | 2.493 | 2.268 | **2.164** | 2.406 | 2.177 |
| 14 | 1.905 | 1.715 | 1.678 | 1.645 | 1.875 | 1.613 | 1.605 | 1.691 | **1.501** |
| 15 | 1.420 | 1.328 | 1.304 | 1.279 | 1.379 | 1.281 | 1.288 | 1.326 | **1.248** |
| 16 | 1.022 | 0.865 | 0.844 | 0.834 | 0.931 | 0.805 | 0.805 | 0.836 | **0.789** |
| 17 | 1.324 | 1.303 | 1.255 | 1.261 | 1.273 | 1.227 | 1.257 | 1.285 | **1.206** |
| 18 | 2.222 | 2.177 | 2.116 | 2.202 | 2.126 | 2.093 | 2.118 | 2.101 | **2.020** |
| 19 | 1.288 | 1.151 | 1.117 | 1.094 | 1.195 | 1.054 | 1.074 | 1.157 | **1.027** |
| 20 | 1.003 | 0.960 | 0.944 | 0.923 | 0.988 | 0.893 | 0.897 | 0.972 | **0.868** |
| 21 | 1.314 | 1.248 | 1.235 | 1.189 | 1.293 | 1.163 | 1.142 | 1.231 | **1.121** |
| 22 | 1.479 | 1.430 | 1.380 | 1.378 | 1.388 | 1.334 | 1.385 | 1.316 | **1.305** |
| 23 | 0.942 | 0.900 | 0.868 | 0.835 | 0.933 | 0.849 | 0.881 | 0.851 | **0.826** |
| 24 | 1.461 | 1.384 | 1.350 | 1.343 | 1.408 | 1.301 | 1.328 | 1.342 | **1.274** |
| Avg | 1.393 | 1.288 | 1.262 | 1.227 | 1.341 | 1.203 | 1.206 | 1.274 | **1.162** |

the proposed solution is lower than AP and LPA, in line with DLMMSE and GBTF, and higher than VCD, LSLCD, and IGD methods. The IGD method, with its emphasis on low complexity, requires between 266 and 374 operations for each 2 by 2 block. Hence, the proposed algorithm offers 0.37 dB of CPSNR performance improvement with about 1.5 to 2 times

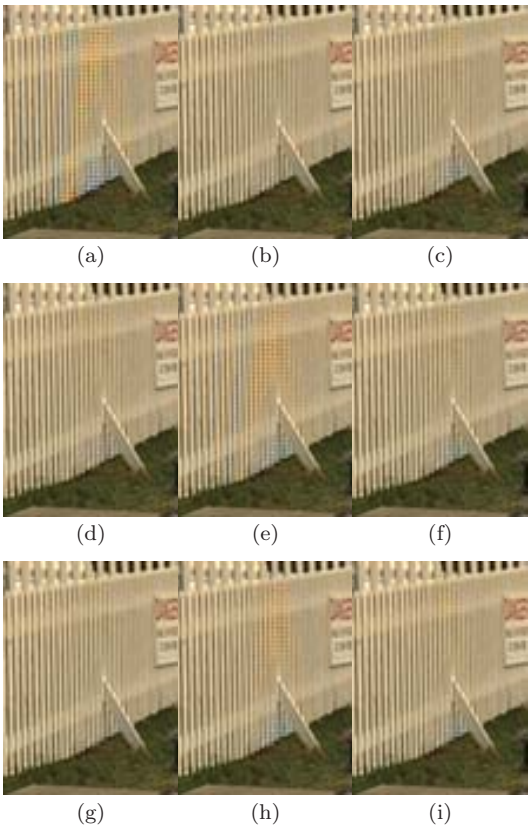


Fig. 5. Comparison on the Bayer pattern. Fence region from the lighthouse image (no. 19). (a) AP. (b) VCD. (c) DLMMSE. (d) LPA. (e) RAD. (f) GBTF. (g) IGD. (h) LSLCD. (i) Proposed.

TABLE III
COMPARISON OF CPSNR VALUES FOR DIFFERENT DEMOSAICING METHODS ON THE LUKAC PATTERN

| No. | QA | AA | ESF | Prop |
|---|---|---|---|---|
| 1 | 36.57 | 37.80 | 39.56 | **39.95** |
| 2 | 38.64 | 39.56 | 40.61 | **40.88** |
| 3 | 40.16 | 41.53 | 42.44 | **42.87** |
| 4 | 39.13 | **40.11** | 39.90 | 40.00 |
| 5 | 35.57 | 36.66 | 36.85 | **37.09** |
| 6 | 37.54 | 39.17 | 40.34 | **41.00** |
| 7 | 40.09 | 41.67 | 41.84 | **42.07** |
| 8 | 35.09 | 36.09 | 37.03 | **37.32** |
| 9 | 40.44 | 41.58 | 42.60 | **42.73** |
| 10 | 40.59 | 41.42 | 41.90 | **42.04** |
| 11 | 38.22 | 39.26 | 40.31 | **40.63** |
| 12 | 41.04 | 42.53 | 43.30 | **43.58** |
| 13 | 34.52 | 34.24 | 35.45 | **35.82** |
| 14 | 34.25 | 35.81 | 36.01 | **36.44** |
| 15 | 38.47 | **39.19** | 39.04 | 39.04 |
| 16 | 40.32 | 42.63 | 43.78 | **44.38** |
| 17 | 40.90 | 41.13 | 41.79 | **41.81** |
| 18 | 36.58 | 36.68 | 37.23 | **37.29** |
| 19 | 38.98 | 39.93 | 41.36 | **41.47** |
| 20 | 39.65 | 40.34 | 41.24 | **41.32** |
| 21 | 37.95 | 38.96 | **40.09** | 40.05 |
| 22 | 37.22 | 37.90 | 38.20 | **38.23** |
| 23 | 40.24 | 41.91 | 42.09 | **42.39** |
| 24 | 35.06 | 35.23 | **35.46** | 35.44 |
| Avg | 38.22 | 39.22 | 39.93 | **40.16** |

the computational complexity of the IGD method. A more detailed complexity analysis for other methods can be found in [22].

TABLE IV
COMPARISON OF S-CIELAB ERROR MEASURE FOR DIFFERENT DEMOSAICING METHODS ON THE LUKAC PATTERN

| No. | QA | AA | ESF | Prop |
|---|---|---|---|---|
| 1 | 1.875 | 1.715 | 1.404 | **1.351** |
| 2 | 1.897 | 1.768 | 1.527 | **1.482** |
| 3 | 1.078 | 0.982 | 0.889 | **0.860** |
| 4 | 1.377 | 1.302 | 1.253 | **1.233** |
| 5 | 2.766 | 2.517 | 2.344 | **2.254** |
| 6 | 1.412 | 1.215 | 1.060 | **0.997** |
| 7 | 1.364 | 1.171 | 1.098 | **1.073** |
| 8 | 1.982 | 1.838 | 1.589 | **1.537** |
| 9 | 0.955 | 0.888 | 0.784 | **0.773** |
| 10 | 0.956 | 0.897 | 0.815 | **0.804** |
| 11 | 1.719 | 1.555 | 1.290 | **1.241** |
| 12 | 0.780 | 0.705 | 0.632 | **0.617** |
| 13 | 2.702 | 2.786 | 2.441 | **2.364** |
| 14 | 2.275 | 2.030 | 1.857 | **1.760** |
| 15 | 1.635 | 1.562 | 1.478 | **1.464** |
| 16 | 1.188 | 0.993 | 0.876 | **0.833** |
| 17 | 1.417 | 1.414 | 1.306 | **1.302** |
| 18 | 2.557 | 2.442 | 2.353 | **2.333** |
| 19 | 1.351 | 1.314 | 1.149 | **1.134** |
| 20 | 1.156 | 1.106 | 0.996 | **0.981** |
| 21 | 1.474 | 1.386 | 1.220 | **1.206** |
| 22 | 1.621 | 1.535 | **1.503** | 1.503 |
| 23 | 1.126 | 1.008 | 0.987 | **0.967** |
| 24 | 1.601 | 1.542 | 1.431 | **1.419** |
| Avg | 1.594 | 1.486 | 1.345 | **1.312** |

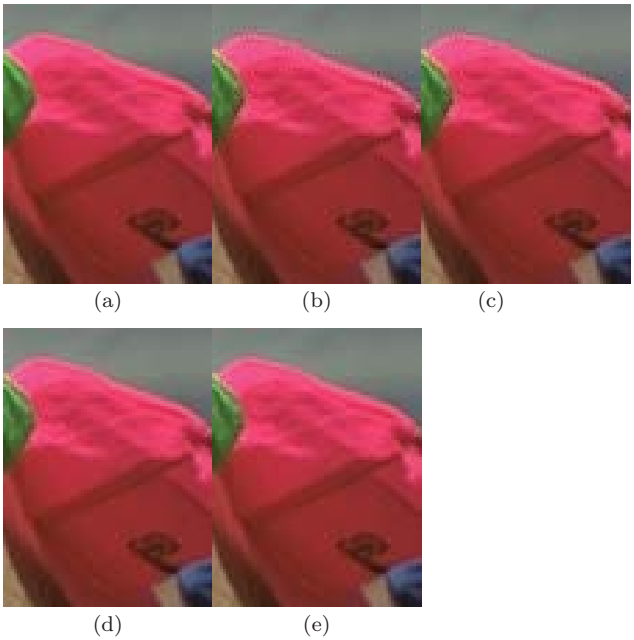


Fig. 6. Comparison on the Lukac pattern. Pink hat from image no. 3. (a) Original. (b) QA. (c) AA. (d) ESF. (e) Proposed.

The simulation results for the Lukac pattern are carried out with the adaptive (AA) and quadratic (QA) approaches proposed in a recent paper on regularization approaches to demosaicing [24], and the Edge Strength Filter (ESF) based method

proposed in [26]. The proposed algorithm gives the best CPSNR for 20 images out of 24. The ESF and AA methods have the best result on 2 images each. The proposed solution outperforms the ESF method by 0.23 dB, and the AA method by 0.94 dB on average. The Lukac pattern CPSNR comparison results are given in Table III, S-CIELAB results are summarized in Table IV, and a sample image region is shown in Figure 6.

## IV. Conclusion

In this paper, we presented a directional CFA interpolation method that is based on multiscale color gradients. The developed method is applied to Bayer and Lukac patterns with great results which shows that the relationship between gradients at different scales can be a very effective feature to optimally combine directional estimates. We believe that the idea behind the proposed method can prove to be useful for image processing problems other than CFA interpolation.

## References


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

[2] R. Lukac and K. N. Plataniotis, "Universal demosaicking for imaging pipelines with an RGB color filter array," *Pattern Recognit.*, vol. 38, no. 11, pp. 2208–2212, Nov. 2005.

[3] K. Hirakawa and P. J. Wolfe, "Spatio-spectral color filter array design for optimal image recovery," *IEEE Trans. Image Process.*, vol. 17, no. 10, pp. 1876–1890, Oct. 2008.

[4] L. Condat, "A new color filter array with optimal properties for noiseless and noisy color image acquisition," *IEEE Trans. Image Process.*, vol. 20, no. 8, pp. 2200–2210, Aug. 2011.

[5] R. Kimmel, "Demosaicing: Image reconstruction from color CCD samples," *IEEE Trans. Image Process*, vol. 8, no. 9, pp. 1221–1228, Sep. 1999.

[6] R. Lukac and K. N. Plataniotis, "A normalized model for color-ratio based demosaicking schemes," in *Proc. Int. Conf. Image Process.*, 2004, pp. 1657–1660.

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

[8] 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, Dec. 1994.

[9] R. H. Hibbard, "Apparatus and method for adaptively interpolating a full color image utilizing luminance gradients," U.S. Patent 5 382 976, Jan. 1995.

[10] J. E. Adams and J. F. Hamilton, "Adaptive color plane interpolation in single color electronic camera," U.S. Patent 5 506 619, Apr. 1996.

[11] K.-H. Chung and Y.-H. Chan, "Color demosaicing using variance of color differences," *IEEE Trans. Image Process.*, vol. 15, no. 10, pp. 2944–2955, Oct. 2006.

[12] L. Zhang and X. Wu, "Color demosaicking via directional linear minimum mean square-error estimation," *IEEE Trans. Image Process.*, vol. 14, no. 12, pp. 2167–2178, Dec. 2005.

[13] D. Paliy, V. Katkovnik, R. Bilcu, S. Alenius, and K. Egiazarian, "Spatially adaptive color filter array interpolation for noiseless and noisy data," *Int. J. Imag. Syst. Technol.*, vol. 17, no. 3, pp. 105–122, 2007.

[14] K. Hirakawa and T. W. Parks, "Adaptive homogeneity-directed demosaicing algorithm," *IEEE Trans. Image Process.*, vol. 14, no. 3, pp. 360–369, Mar. 2005.

[15] D. Menon, S. Andriani, and G. Calvagno, "Demosaicing with directional filtering and a posteriori decision," *IEEE Trans. Image Process.*, vol. 16, no. 1, pp. 132–141, Jan. 2007.

[16] J. W. Glotzbach, R. W. Schafer, and K. Illgner, "A method of color filter array interpolation with alias cancellation properties," in *Proc. IEEE Int. Conf. Image Process.*, vol. 1. 2001, pp. 141–144.

[17] B. K. Gunturk, Y. Altunbasak, and R. M. Mersereau, "Color plane interpolation using alternating projections," *IEEE Trans. Image Process.*, vol. 11, no. 9, pp. 997–1013, Sep. 2002.

[18] N.-X. Lian, L. Chang, Y.-P. Tan, and V. Zagorodnov, "Adaptive filtering for color filter array demosaicking," *IEEE Trans. Image Process.*, vol. 16, no. 10, pp. 2515–2525, Oct. 2007.

[19] X. Li, B. Gunturk, and L. Zhang, "Image demosaicing: A systematic survey," *Proc. SPIE*, vol. 6822, pp. 68221J-1–68221J-15, Jan. 2008.

[20] D. Alleysson, B. C. Lavaréne, S. Süsstrunk, and J. Hérault, "Linear minimum mean square error," in *Single-Sensor Imaging: Methods and Applications for Digital Cameras*. Boca Raton, FL: CRC Press.

[21] D. Menon, "Color image reconstruction for digital cameras," Ph.D. dissertation, Dept. Inform. Eng., Univ. Padova, Padova, Italy, 2008.

[22] K. H. Chung and Y. H. Chan, "Low-complexity color demosaicing algorithm based on integrated gradients," *J. Electron. Imag.*, vol. 19, no. 2, pp. 021104-1–021104-15, Jun. 2010.

[23] I. Pekkucuksen and Y. Altunbasak, "Gradient based threshold free color filter array interpolation," in *Proc. IEEE Int. Conf. Image Process.*, Sep. 2010, pp. 137–140.

[24] D. Menon and G. Calvagno, "Regularization approaches to demosaicking," *IEEE Trans. Image Process.*, vol. 18, no. 10, pp. 2209–2220, Oct. 2009.

[25] B. Leung, G. Jeon, and E. Dubois, "Least-squares luma-chroma demultiplexing algorithm for bayer demosaicking," *IEEE Trans. Image Process.*, vol. 20, no. 7, pp. 1885–1894, Jul. 2011.

[26] I. Pekkucuksen and Y. Altunbasak, "Edge oriented directional color filter array interpolation," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, May 2011, pp. 993–996.


**Ibrahim Pekkucuksen** (M'12) received the B.S. degree from Texas A&M University, College Station, in 2005, and the M.S. and Ph.D. degrees from the Georgia Institute of Technology, Atlanta, in 2007 and 2011, respectively, all in electrical engineering.

He is currently a Technical Staff Member with the Texas Instruments Systems and Applications R&D Center, Dallas. His current research interests include resolution enhancement, demosaicing, and computational photography.

**Yucel Altunbasak** (SM'02–F'12) received the M.S. and Ph.D. degrees in electrical and computer engineering from the University of Rochester, New York, NY.

He joined Hewlett-Packard Research Laboratories, Palo Alto, CA, in 1996. He was a Consulting Assistant Professor with Stanford University, Stanford, CA, and San Jose State University, San Jose, CA.

Dr. Altunbasak served as the Technical Program Chair of the International Consortium for Intergenerational Programs in 2006. He was an Associate Editor of the IEEE Transactions on Image Processing, the IEEE Transactions on Signal Processing, *Signal Processing: Image Communication*, and the *Journal of Circuits, Systems and Signal Processing*. He served as a Lead Guest Editor of two *Signal Processing: Image Communication* special issues on wireless video and video networking. He was a Guest Editor of the IEEE Journal on Selected Topics on Signal Processing's special issue on Network-Aware Multimedia Processing and Communications. He was the Vice President of the IEEE Communications Society Multimedia Communications' Technical Committee. He has been elected for the Technical Committees of the IEEE Signal Processing Society Image and Multidimensional Signal Processing, Multimedia Signal Processing, and Bio Imaging and Signal Processing. He was a Co-Chair of the Advanced Signal Processing for Communications Symposia at ICC in 2003. He was the Track Chair of ICME in 2003 and ICME in 2004, the Panel Sessions Chair of ITRE in 2003, and the Session Chair of various international conferences. He has co-authored the paper that received the Most Cited Paper Award from the *EURASIP Journal of Signal Processing: Image Communication* in 2008. He is the co-author of two conference papers that received the Best Student Paper Award from ICIP in 2003 and VCIP in 2006. He co-authored a conference paper that was selected as a design finalist at EMBS in 2004. He was a recipient of the U.S. National Science Foundation CAREER Award in 2002 and the Outstanding Junior Faculty Member Award from the School of Electrical and Computer Engineering, Georgia Institute of Technology, in 2003.