

Demosaicking: Color Filter Array Interpolation in Single-Chip Digital Cameras

B. K. Gunturk, J. Glotzbach, Y. Altunbasak, R. W. Schafer, and R. M. Mersereau¹

Draft for the IEEE SPM Special Issue on Color Image Processing

I. INTRODUCTION

Digital cameras have become popular and many people are choosing to take their pictures with digital cameras instead of film cameras. When a digital image is recorded, the camera needs to perform a significant amount of processing to provide the user a viewable image. This processing includes white balance adjustment, gamma correction, compression and more (reference “Color Image Processing Pipeline in Digital Still Cameras” in this issue). A very important part of this image processing chain is color filter array interpolation or demosaicking.

A color image requires at least three color samples at each pixel location. Computer images often use red, green, and blue. A camera would need three separate sensors to completely measure the image. Using multiple sensors to detect different parts of the visible spectrum requires splitting the light entering the camera so that the scene is imaged onto each sensor. Precise registration is then required to align the three images. These additional requirements add a large expense to the system. Thus, many cameras use a single sensor array with a color filter array. The color filter array allows only one part of the spectrum to pass to the sensor so that only one color is measured at each pixel. This means that the camera must estimate the missing two color values at each pixel. This process is known as demosaicking.

Several patterns exist for the filter array. The most common array is the Bayer color filter array, shown in Figure 1. The Bayer array measures the green image on a quincunx grid and the red and blue images on rectangular grids. The green image is measured at a higher sampling rate because the peak sensitivity of the human visual system lies in the medium wavelengths, corresponding to the green portion of the spectrum.

Other patterns exist - the Nikon Coolpix 990 uses a CMYG grid, where each of the four images (cyan, magenta, yellow, green) are sampled using rectangular grids. A CMY-based system has the advantage of

¹B. K. Gunturk is with the Louisiana State University; J. Glotzbach, Y. Altunbasak, R. W. Schafer, and R. M. Mersereau are with the Georgia Institute of Technology. Email: bahadir@ece.lsu.edu, glotz@ece.gatech.edu, yucel@ece.gatech.edu, rws@ece.gatech.edu, rmm@ece.gatech.edu. This work was supported in part by Texas Instruments Leadership University Program, ONR N00014-01-1-0619, and NSF CCR-0113681.

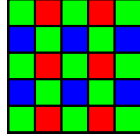


Fig. 1. Bayer color filter array arrangement.

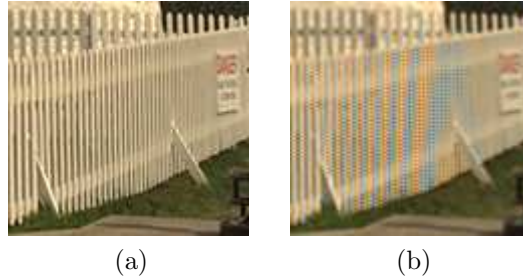


Fig. 2. Bicubic interpolation used for color filter array interpolation results in noticeable artifacts. (a) Original image. (b) Bicubic interpolation.

being more sensitive to light because the incoming light only has to pass through one layer of filters. Red, green, and blue filters are generated by overlaying combinations of cyan, magenta, and yellow filters [2]. For example, the combination of cyan and magenta filters would make a blue filter. Even though other options exist, this article discusses the demosaicking problem with reference to the Bayer RGB color filter array.

If the measured image is divided by measured color into three separate images, this problem looks like a typical image interpolation problem. Therefore, one might try to apply standard image interpolation techniques. Bi-cubic interpolation is a common image interpolation technique that produces good interpolation results when applied to grayscale images. However, when bi-cubic interpolation is used for this problem, the resulting image shows many visible artifacts. This is illustrated in Figure 2.

This result motivates the need to find a specialized algorithm for the demosaicking problem. Bi-cubic interpolation and other standard interpolation techniques treat the color image as three independent images. However, the three images in a color image are generally highly correlated. Many algorithms have been published suggesting how to use this correlation. This article surveys many of these algorithms and discusses the results in terms of objective and subjective measures.

II. IMAGE FORMATION PROCESS

Since some of the demosaicking methods make explicit use of image formation models, we provide a brief summary of image formation before reviewing the demosaicking methods. The imaging process is usually modeled as a linear process between the light radiance arriving at the camera and the pixel intensities produced by the sensors. Most digital cameras use charge-coupled device (CCD) sensors. In a CCD camera, there is a rectangular grid of electron-collection sites laid over a silicon wafer to record the amount of light energy reaching each of them. When photons strike these sensor sites, electron-hole pairs are generated; and the electrons generated at each site are collected over a certain period of time. The numbers of electrons are eventually converted to pixel values.

Each sensor type, S , has a specific spectral response $L_S(\lambda)$, which is a function of the spectral wavelength λ , and a spatial response $h_S(x, y)$, which results from optical blur and spatial integration at each sensor site. In practice, discrete formulation of the imaging process is used:

$$S(n_1, n_2) = \sum_l \sum_{m_1, m_2} L_S(l) h_S(n_1 - m_1, n_2 - m_2) r(m_1, m_2, l) + N_S(n_1, n_2). \quad (1)$$

where $S(n_1, n_2)$ is the intensity at spatial location (n_1, n_2) , $r(m_1, m_2, l)$ is the incident radiance, and $N_S(n_1, n_2)$ is the additive noise that is a result of thermal/quantum effects and quantization. There are a couple of assumptions in this formulation: (i) the input-output relation is assumed to be linear; (ii) the spatial blur $h_S(n_1, n_2)$ is assumed to be space-invariant and independent of wavelength; (iii) only the additive noise is considered. These assumptions are reasonable for practical purposes.

The last step in the imaging process is the color filter array (CFA) sampling. Denoting Λ_S as the set of pixel locations, (n_1, n_2) , for channel S , a CFA mask function can be defined as

$$f_S(n_1, n_2) = \begin{cases} 1, & (n_1, n_2) \in \Lambda_S \\ 0, & \text{otherwise} \end{cases}. \quad (2)$$

In the Bayer CFA, there are three types of color channels: red (R), green (G), and blue (B). Therefore, for the Bayer CFA, the observed data, $O(n_1, n_2)$, is

$$O(n_1, n_2) = \sum_{S=R,G,B} f_S(n_1, n_2) S(n_1, n_2). \quad (3)$$

III. DEMOSAICKING METHODS

We examine demosaicking methods in three groups. The first group consists of heuristic approaches. The second group formulates demosaicking as a restoration problem. The third group is a generalization that uses the spectral filtering model given in (1) in restoration.

A. Group I: Heuristic Approaches

Heuristic approaches do not try to solve a mathematically defined optimization problem. They are mostly filtering operations that are based on reasonable assumptions about color images. Heuristic approaches may be spatially adaptive, and they may exploit correlation among the color channels. We now present these heuristic approaches.

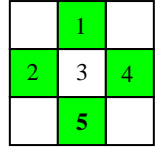
1) *Edge-Directed Interpolation*: Although non-adaptive algorithms (e.g., bilinear interpolation, bicubic interpolation) can provide satisfactory results in smooth regions of an image, they usually fail in textured regions and edges. *Edge-directed interpolation* is an adaptive approach, where the area around each pixel is analyzed to determine if a preferred interpolation direction exists. In practice, the interpolation direction is chosen to avoid interpolating across edges, instead interpolating along any edges in the image.

An illustration of edge-directed interpolation is shown in Figure 3, where horizontal and vertical gradients at a missing green location are calculated from the adjacent green pixels. In [17], these gradients are compared to a constant threshold. If the gradient from one direction falls below the threshold, interpolation is performed only along this direction. If both gradients are below the threshold or both gradients are above the threshold, the pixels along both directions are used to estimate the missing value.

The edge-directed interpolation idea can be modified by using larger regions (around the pixel in question) with more complex predictors and by exploiting the texture similarity in different color channels. In [23], the red and blue channels (in the 5×5 neighborhood of the missing pixel) are used instead of the green channel to determine the gradients. To determine the horizontal and vertical gradients at a blue (red) sample, second-order derivatives of blue (red) values are used. This algorithm is illustrated in Figure 4.

Another example of the edge-directed interpolation is found in [19], where the Jacobian of the red, green, and blue samples is used to determine edge directions.

2) *Constant-Hue-Based Interpolation*: One commonly used assumption in demosaicking is that the hue (color ratios) within an object in an image is constant. In [22], this is explained that an object of constant color will have a constant color ratio even though lighting variations may change the measured values. This perfect inter-channel correlation assumption is sometimes formulated such that the color differences (or logarithm of color ratios) within objects are constant. This constant color ratio (or difference) assumption prevents abrupt changes in color intensities, and has been extensively used for the interpolation of the chrominance (red and blue) channels [9], [34], [3], [23], [17], [10], [22], [30], [27]. The demosaicking algorithms that are based on this assumption are called *constant-hue-based interpolation* or *smooth-hue*



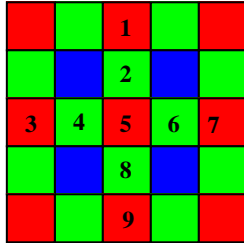
1. Calculate horizontal gradient $\Delta H = |G2 - G4|$
2. Calculate vertical gradient $\Delta V = |G1 - G5|$
3. If $\Delta H > \Delta V$,

$$G3 = (G1 + G5)/2$$
 Else if $\Delta H < \Delta V$,

$$G3 = (G2 + G4)/2$$
 Else

$$G3 = (G1 + G5 + G2 + G4)/4$$

Fig. 3. Edge-directed interpolation for green channel is illustrated. $G1$, $G2$, $G4$, and $G5$ are measured green values; $G3$ is the estimated green value at pixel 3.



1. Calculate horizontal gradient $\Delta H = |(R3 + R7)/2 - R5|$
2. Calculate vertical gradient $\Delta V = |(R1 + R9)/2 - R5|$
3. If $\Delta H > \Delta V$,

$$G5 = (G2 + G8)/2$$
 Else if $\Delta H < \Delta V$,

$$G5 = (G4 + G6)/2$$
 Else

$$G5 = (G2 + G8 + G4 + G6)/4$$

Fig. 4. Edge-directed interpolation in [23] is illustrated for estimating the green (G) value at pixel 5. The red (R) values are used to determine the edge direction. When the missing green pixel is at a blue pixel, the blue values are used to determine the edge direction.

transition methods.

As a first step, these algorithms interpolate the luminance (green) channel, which is done using bilinear or edge-directed interpolation. The chrominance (red and blue) channels are then estimated from the interpolated “red hue” (red-to-green ratio) and “blue hue” (blue-to-green ratio). To be more explicit, the interpolated “red hue” and “blue hue” values are multiplied by the green value to determine the missing red and blue values at a particular pixel location. The hues can be interpolated with any method (bilinear, bi-cubic, edge-directed, etc.). Instead of interpolating the color ratios, the color differences can also be interpolated, as described in Figure 5.

3) *Weighted Average:* In edge-directed interpolation, the edge direction is found first, and then the missing sample is estimated by interpolating along the edge. Instead, the likelihood of an edge in a certain direction can be found, and the interpolation can be done based on the edge likelihoods. Such an algorithm was proposed by Kimmel in [22]. The algorithm defines edge indicators in several directions as measures of edge likelihood in those directions, and determines a missing pixel intensity as a weighted sum of its neighbors. If the likelihood of an edge crossing in a particular direction is high, the edge indicator returns

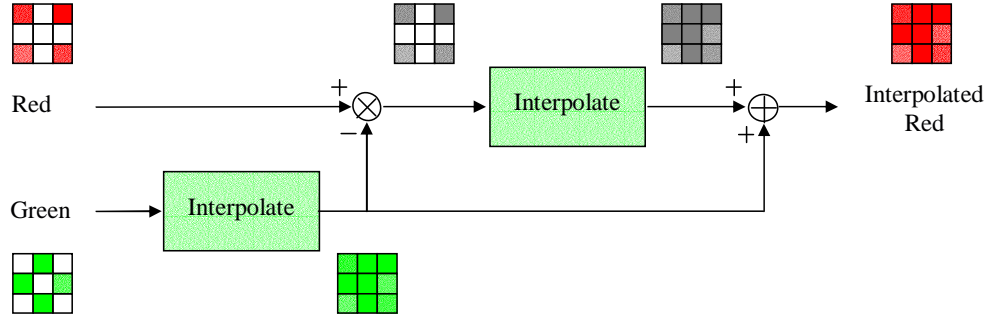


Fig. 5. Constant-difference-based interpolation is illustrated for red channel. Blue channel is interpolated similarly.

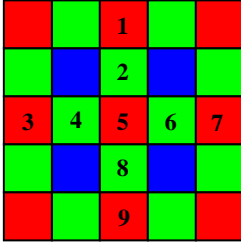
a small value, which results in less contribution from the neighboring pixel in that direction. The green channel is interpolated first; the red and blue channels are interpolated from the red/green and blue/green ratios. The color channels are then updated iteratively to obey the constant color ratio rule.

A similar algorithm was proposed recently in [25], where edge indicators are determined in a 7×7 window for the green and a 5×5 window for the red/blue channels. In this case, the edge indicator function is based on the L_1 norm (absolute difference) as opposed to the L_2 norm of [22]. A related algorithm is proposed in [35], where the directions (horizontal, vertical, diagonal) that have the smallest two gradients are used in interpolation.

A different example of weighted directional interpolation can be found in [33], where fuzzy membership assignment is used to compute weights for the horizontal and vertical direction. The weights are computed experimentally and used as constants in the algorithm.

4) *Second-Order Gradients As Correction Terms* : In [4], Hamilton and Adams begin by using edge-directed interpolation for the green image. Correction terms from the red and blue samples are added to this initial estimate. They compute the Laplacian for the red or blue samples along the interpolation row or column and use this to correct the simple averaging interpolation. This correction term reduces aliasing passed to the output by the simple averaging filter. Figure 6 illustrates this algorithm.

5) *Alias Canceling Interpolation*: In [12], the green image is used to add high-frequency information and reduce aliasing in the red and blue images. First, the red and blue images are interpolated with a rectangular lowpass filter according to the rectangular sampling grid. This fills in the missing values in the grid, but allows aliasing distortions into the red and blue output images. These output images are also missing the high-frequency components needed to produce a sharp image. However, because the green image is sampled at a higher rate, the high-frequency information can be taken from the green



1. Calculate horizontal gradient $\Delta H = |G4 - G6| + |R5 - R3 + R5 - R7|$
2. Calculate vertical gradient $\Delta V = |G2 - G8| + |R5 - R1 + R5 - R9|$
3. If $\Delta H > \Delta V$,

$$G5 = (G2 + G8)/2 + (R5 - R1 + R5 - R9)/4$$
 Else if $\Delta H < \Delta V$,

$$G5 = (G4 + G6)/2 + (R5 - R3 + R5 - R7)/4$$
 Else

$$G5 = (G2 + G8 + G4 + G6)/4 + (R5 - R1 + R5 - R9 + R5 - R3 + R5 - R7)/8$$

Fig. 6. The Hamilton and Adams method [4] is illustrated for estimating the green (G) value at pixel 5. The red (R) and green values are used to determine the edge direction and to estimate the missing value. When the missing green pixel is at a blue pixel, the blue and green values are used.

image to improve an initial interpolation of the red and blue images. A horizontal highpass filter and a vertical highpass filter are applied to the green image. This provides the high-frequency information that the low sampling rate of the red and blue images cannot preserve. Aliasing occurs when high-frequency components are shifted into the low-frequency portion of the spectrum, so if the outputs of the highpass filters are modulated into the low-frequency regions, an estimate of the aliasing in the red and blue images can be found. This estimate is used to reduce the aliasing in the red and blue images, as illustrated in Figure 7. This method relies on the assumption that the high-frequency information in the red, green, and blue images is identical. If this assumption does not hold, the addition of the green information into the red and blue images can add unwanted distortions. This method also makes the assumption that the input image is band-limited within the diamond-shaped Nyquist region of the green quincunx sampling grid. When this assumption fails, the aliasing artifacts are enhanced instead of reduced because the green image also contains aliasing.

6) *Homogeneity-Directed Interpolation*: Instead of choosing the interpolation direction based on edge indicators, it is possible to use different measures. In [18], local homogeneity is used as an indicator to choose between horizontally and vertically interpolated intensities. The homogeneity-directed interpolation imposes the similarity of the luminance and chrominance values within small neighborhoods. The RGB data is first interpolated horizontally and vertically. That is, there are two candidates for each missing color sample. The decision for which one to choose is made in the CIELab space. Both the horizontally and vertically interpolated images are transformed to the CIELab space. In the CIELab space, either the horizontally or the vertically interpolated pixel values are chosen based on the local homogeneity. The local homogeneity is measured by the total number of *similar* luminance and chrominance values of the pixels that are within a neighborhood of the pixel in question. Two values are taken as *similar* when the

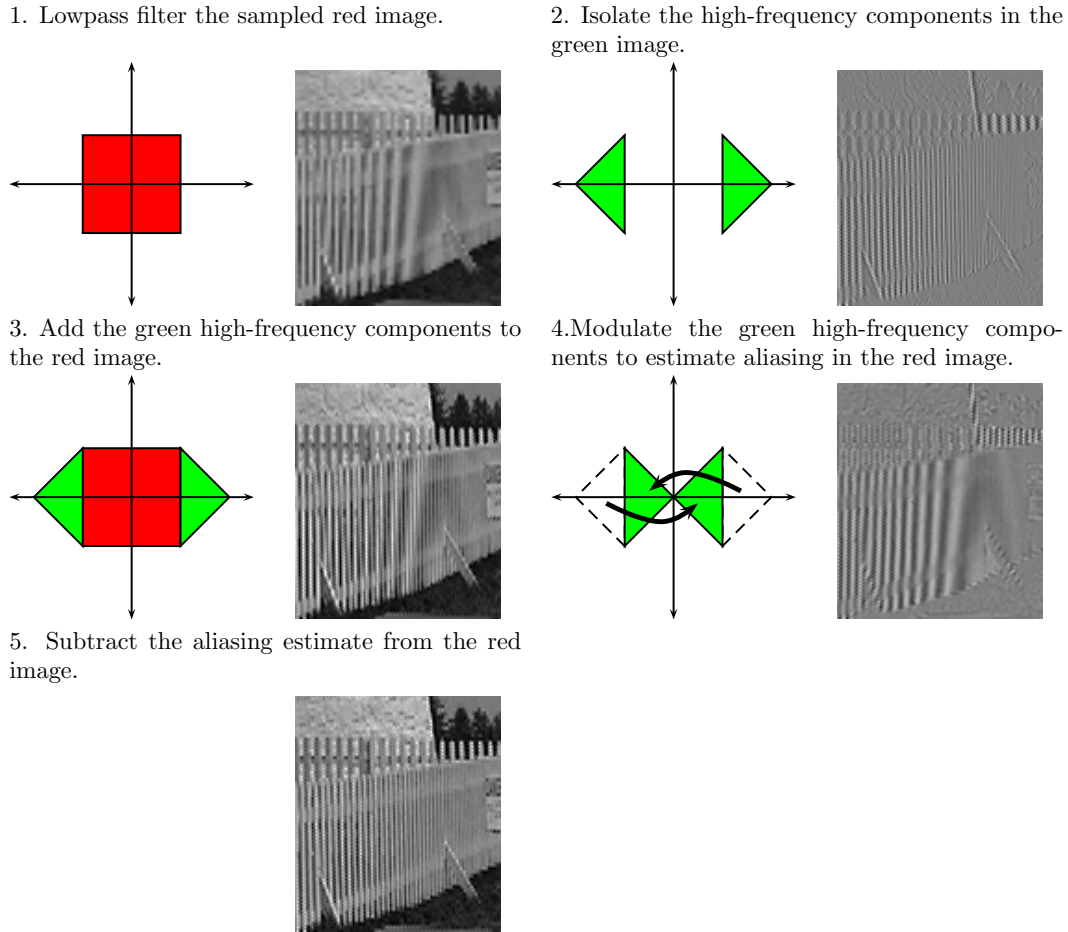


Fig. 7. High-frequency information from the green image is modulated and used to cancel aliasing in the red image.

Euclidian distance between them is less than a threshold.

7) *Pattern Matching*: Several algorithms attempt to find a pattern in the data or fit the data to one of several templates. A different interpolator is applied for each template. This allows a different method to be used for edges and smooth regions.

In [9], Cok describes a pattern matching algorithm to be used on the green image. Each missing green value is classified as a stripe, edge, or corner, corresponding to the features expected to be found in natural images. After classifying the pixel, an appropriate interpolator is applied to estimate the missing value.

In [8], Chang *et al.* introduce a method using directional information and added the ability to use multiple directions. This method uses eight possible horizontal, vertical, and diagonal interpolation directions. A gradient is computed for each direction and then a threshold is computed based on these

gradients to determine which directions are used. For each direction included in the interpolation, an average red, green, and blue value is computed. For each of the missing colors at the current pixel, the difference between the average of the missing color and the average of the color of the current pixel is calculated. This color difference is added to the value of the current pixel to estimate the missing color value.

In [28], bilateral filtering is used to combine a standard interpolation filter with the local properties of the image. A goal of the bilateral filter could be to enforce color similarity between neighboring pixels. This filtering approach allows for both smoothing of noise and sharpening of edges.

8) *Vector-Based Interpolation*: In this approach, each pixel is considered as a vector in the color space, and interpolation is designed to minimize the angle or the distance among the neighboring vectors. One of the algorithms proposed in [21] is based on the minimization of angles in spherical coordinates. After an initial interpolation of missing samples, each pixel is transformed to spherical coordinates, (ρ, θ, ϕ) . The relationship between the (R, G, B) space and (ρ, θ, ϕ) space is

$$R = \rho \cos(\theta) \sin(\phi); \quad G = \rho \cos(\theta) \cos(\phi); \quad \text{and} \quad B = \rho \sin(\theta). \quad (4)$$

In the (ρ, θ, ϕ) space, some filtering operation, such as median filtering, is applied to the angles θ and ϕ only. This forces the chrominance components to be similar. Because ρ is closely related to the luminance component, keeping it unchanged preserves the luminance discontinuities among neighboring pixels. After the filtering process, the image is transformed back to the (R, G, B) space, and original measured samples are inserted into their corresponding locations. Spherical domain filtering and insertion operations are repeated iteratively.

Another vector-based interpolation is proposed in [15]. In contrast to the approach in [21], the RGB vectors are constructed from observed data only. All possible red, green, and blue combinations in a 3×3 neighborhood of a pixel are used to form the so-called pseudo-pixels. The colors at the center of the 3×3 region are found from the vector median of the pseudo-pixels. The formation of pseudo-pixels is illustrated in Figure 8. The vector median (*VM*) operation is defined as

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = VM \left\{ \begin{bmatrix} v_{11} \\ v_{12} \\ v_{13} \end{bmatrix}, \begin{bmatrix} v_{21} \\ v_{22} \\ v_{23} \end{bmatrix}, \dots, \begin{bmatrix} v_{N1} \\ v_{N2} \\ v_{N3} \end{bmatrix} \right\} \equiv \arg \min_{x_1, x_2, x_3} \sum_{i=1}^N \left(\sum_{k=1}^3 (x_k - v_{ik})^2 \right)^{1/2}. \quad (5)$$

There is no closed form solution to (5); the vector median can be found iteratively by some numerical methods [15]. Note that the reconstructed color channels are not necessarily consistent with the observed data. It is argued that this would reduce color artifacts even if the edge locations are shifted in the end.

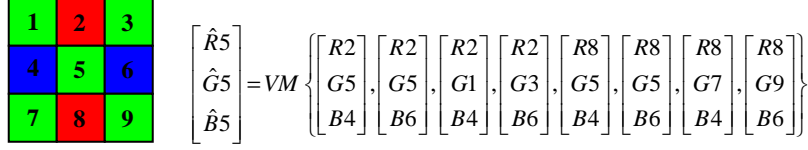


Fig. 8. The formation of pseudo-pixels in [15] is shown. Vector median (VM) operation is applied to the pseudo-pixels to estimate the colors at pixel 5.

9) *Fourier-Domain Filtering*: In [5], it is shown that CFA samples can be written as a summation of luminance and chrominance terms, which are well localized in frequency domain. Therefore, the luminance and chrominance terms can be recovered by lowpass and highpass filtering. The formulation starts with the representation of a CFA data, $O(n_1, n_2)$, in terms of red, green, and blue channels:

$$O(n_1, n_2) = \sum_{S=R,G,B} m_S(n_1, n_2) S(n_1, n_2), \quad (6)$$

where $m_S(n_1, n_2)$ are the modulation functions defined as

$$m_R(n_1, n_2) = (1 + \cos(\pi n_1)) (1 + \cos(\pi n_2)) / 4, \quad (7)$$

$$m_G(n_1, n_2) = (1 - \cos(\pi n_1)) (1 + \cos(\pi n_2)) / 2, \quad (8)$$

$$m_B(n_1, n_2) = (1 - \cos(\pi n_1)) (1 - \cos(\pi n_2)) / 4. \quad (9)$$

Using the definitions of the modulation functions, (6) can be written as the summation two terms:

$$O(n_1, n_2) = \frac{1}{4} (R(n_1, n_2) + 2G(n_1, n_2) + B(n_1, n_2)) + \sum_{S=R,G,B} \tilde{m}_S(n_1, n_2) S(n_1, n_2). \quad (10)$$

The first term in (10) is called as the luminance term because it does not depend on the modulation functions; the second term is called the chrominance term. In the Fourier domain, the luminance terms are located in the low-frequency regions, while the chrominance terms are located in the high-frequency regions. Although there may be some spectra overlap, the luminance and chrominance can be estimated by low-pass filtering and high-pass filtering, respectively. Red, green, and blue samples are found using the luminance and chrominance terms.

B. Group II: Reconstruction Approaches

The second group of algorithms makes some assumptions about the inter-channel correlation or the prior image, and solves a mathematical problem based on those assumptions. One of the methods proposed in [21] uses spatial smoothness and color correlation terms in a cost function that is minimized iteratively.

In [14], an iterative algorithm that forces similar high frequency components among the color channels, and ensures data consistency is proposed. In [26], demosaicking problem is formulated as a Bayesian estimation problem, where spatial smoothness and constant hue assumptions are used as regularization terms.

1) *Regularization*: In [21], Keren and Osadchy propose a regularization approach, which minimizes a cost function consisting of a spatial smoothness term and a color correlation term. To write the cost function, we first define vector $V(n_1, n_2)$ as

$$V(n_1, n_2) = [R(n_1, n_2) - \bar{R}, G(n_1, n_2) - \bar{G}, B(n_1, n_2) - \bar{B}]^T, \quad (11)$$

where \bar{R} , \bar{G} , and \bar{B} are the average colors in the vicinity of the pixel at (n_1, n_2) . Denoting $C_{n_1 n_2}$ as the covariance matrix of the RGB values, and $S_{n_1 n_1}$, $S_{n_1 n_2}$, and $S_{n_2 n_2}$ as the spatial derivatives in the horizontal, diagonal, and vertical direction, respectively, the cost function is defined as

$$Cost = \int \int \sum_{S=R,G,B} (S_{n_1 n_1}^2 + 2S_{n_1 n_2}^2 + S_{n_2 n_2}^2) dn_1 dn_2 + \lambda \int \int V(n_1, n_2)^T C_{n_1 n_2}^{-1} V(n_1, n_2) dn_1 dn_2, \quad (12)$$

where λ is a positive constant. Restoration is achieved by minimizing this cost function iteratively. The algorithm starts with an initial interpolation of the missing values, estimates the local averages and covariance matrix based on the current values, and minimizes the cost function using a finite-element method. In another version of the algorithm, the second term in (12) is replaced by the summation of the squared norms of the vector products of neighboring pixels. Since the vector product gives sine of the angle between two vectors, this term tries to minimize the angle among neighboring pixels.

2) *Projections onto Convex Sets (POCS) Approach*: In [14], Gunturk *et al.* propose an algorithm that enforces similar high-frequency characteristics for the red, green, and blue channels, and ensures that the resulting image is consistent with the observed data. The algorithm defines two constraint sets, and reconstructs the color channels using a projections onto convex sets (POCS) technique. The ‘‘observation’’ constraint set ensures that the interpolated color channels are consistent with the observed data. That is, the color samples captured by the digital camera are not changed during the reconstruction process. The ‘‘detail’’ constraint set imposes similar high-frequency components in the color channels. The formal definition of the ‘‘detail’’ constraint set is based on the subband decomposition of the color channels: The absolute difference between the detail subbands of the red (blue) channel and the green channel is constrained to be less than a threshold at each spatial location. These two constraint sets are shown to be convex in [14]. According to the algorithm, the color channels are first interpolated to get the initial estimates. Red and blue channels are then updated by projection onto the ‘‘detail’’ constraint set

and the “observation” constraint set iteratively. Projection onto “detail” constraint set is performed by (i) decomposing the color channels into frequency subbands with a bank of analysis filters, (ii) updating the detail subbands of the red and blue channels so that they are within a threshold distance to the detail subbands of the green channel, and (iii) restoring them with a bank of synthesis filters. Projection onto the “observation” constraint set is performed by inserting the observed data into their corresponding locations in the color channels.

3) *Bayesian Approach*: With the Bayesian estimation approach, it is possible to incorporate prior knowledge about the solution (such as spatial smoothness and constant color ratio) and the noise statistics into the solution. In the maximum *a posteriori* probability (MAP) formulation, the observed data $O(n_1, n_2)$, the full color channels $S(n_1, n_2)$, and the additive noise $N_S(n_1, n_2)$ are all assumed to be random processes. Denoting $p(S|O)$ as the conditional probability density function (PDF), the MAP estimate \hat{S} is given by

$$\hat{S} = \arg \max_S \{p(S|O)\} = \arg \max_S \{p(O|S)p(S)\}, \quad (13)$$

where Bayes’ rule and the fact that O is independent of S were used. To find the MAP estimate \hat{S} , the conditional PDF, $p(O|S)$, and the prior PDF, $p(S)$, need to be modeled. The conditional PDF $p(O|S)$ is derived from the noise statistics, which is usually assumed to be white Gaussian. As for the prior PDF, different models have been proposed.

In [26] and [16], Markov Random Field (MRF) models were used. In MRF processing, the conditional and prior PDFs can be modeled as Gibbs distributions. The Gibbs distribution has an exponential form, and it is characterized by an *energy function* and a *temperature* parameter. A PDF with Gibbs distribution can be written as

$$p(x) = \frac{1}{Z} e^{-U(x)/T}, \quad (14)$$

where $U(\cdot)$ is the energy function, T is the temperature parameter, and Z is the normalization constant. One feature of the MRF is that the total energy function U can be written as a sum of local energy functions, which allows for localized reconstruction [11]. In [26], three types of local energy functions are defined at each pixel location. The first energy function is associated with the additive noise, the second energy function imposes spatial smoothness, and the third energy function imposes constancy of cross color ratios. Once the local energy functions are defined, the solution minimizing the total energy can be found using a variety of techniques. In [26], *simulated annealing* technique was used.

As an alternative, [16] proposes a prior based on the steerable wavelet decomposition. With the steerable wavelet decomposition, images can be represented as a sum of bandpass components, each of which can

be decomposed into a set of oriented bands using steerable filters. Such a directional decomposition enables imposing edge-oriented smoothness instead of an isotropic smoothness. Therefore, across-the-edge averaging is avoided. Directional energy functions are defined at different scales of a Laplacian pyramid; and a gradient descent procedure is applied to find the image that minimizes the energy functions at all scales.

With prior knowledge that a compression block is likely to follow the interpolation process, [6] proposes a Bayesian approach where the distribution of the compression coefficients is assumed and used to select the best demosaicked image. This provides a framework where the demosaicking process uses prior knowledge of the compression process.

4) *Artificial Neural Network Approach:* Demosaicking is an under-determined problem. Assumptions such as spatial smoothness and constant hue are used to regularize it. Obviously, these assumptions are not necessarily correct for all cases. The use of artificial neural networks is an alternative approach. The artificial neural network (ANN) approach uses training images to learn the parameters to be used in reconstruction. In [20], three methods based on Artificial Neural Networks are proposed: Perceptron, Backpropagation, and Quadratic Perceptron. In all these methods, images are processed in 2×2 regions of the Bayer CFA. To have more information about the local characteristics, pixels around the 2×2 regions are also used as inputs. That is, 16 inputs are supplied to the network, and eight outputs (two missing values at each 2×2 region) are estimated. In the Perceptron method, the outputs are linear combinations of the inputs. The weights are learned from training data. It turns out that the Perceptron network is not satisfactory in high-frequency regions. The Backpropagation network is capable of learning complex nonlinear functions, and produces better results in the high frequency regions than the Perceptron network. On the other hand, the Backpropagation network fails in the low frequency regions due to the nonlinearity of the sigmoid function used in the Backpropagation process. To solve this problem, Kapah and Hel-Or proposes a selector, which is also an ANN, to select either the output of the Perceptron network or the Backpropagation network at each 2×2 region. The last method is the Quadratic Perceptron network. In contrast the Perceptron network, the weights are not fixed but functions of inputs. An additional Perceptron subnetwork is used to produce the weights. The overall performance of the Quadratic Perceptron network is reported to be the best in [20]. Another algorithm based on ANN is proposed in [13], where three-layer feedforward structure is used. For each color channel, sixteen measured pixels around a missing pixel are used as the input.

C. Group III: Image Formation Modeling

The last group of methods uses a model of the image formation process and formulates the demosaicking problem as an inverse problem. These algorithms account for the transformations performed by the color filters, lens distortions, sensor noise, etc. and determine the most likely output image given the measured CFA image. Referring to equations (1) and (3), the purpose is to reconstruct the radiance $r(m_1, m_2, l)$. Defining \mathbf{r} , \mathbf{O} , and \mathbf{N} as the stacked forms of $r(m_1, m_2, l)$, $O(n_1, n_2)$, and $N_S(n_1, n_2)$, respectively; the observation model can be written in the compact form

$$\mathbf{O} = \mathbf{H}\mathbf{r} + \mathbf{N}, \quad (15)$$

where \mathbf{H} is the matrix that includes the combined effects of optical blur, sensor blur, spectral response, and CFA sampling. In [31], [32], and [7] the minimum mean square error (MMSE) solution of (3) is given:

$$\hat{\mathbf{r}} = E[\mathbf{r}\mathbf{O}^T] \left(E[\mathbf{O}\mathbf{O}^T] \right)^{-1} \mathbf{O}, \quad (16)$$

where $E[\cdot]$ is the expectation operation. In [7], the point spread function is taken as an impulse function; and \mathbf{r} is represented as a weighted sum of spectral basis functions to reduce the dimensionality of the problem. ([24] provides examples of [7] with a PSF included in the reconstruction.) In [32], adaptive reconstruction and ways to reduce computational complexity are discussed. In [31], a finite-support filter is derived based on the assumption that the radiance, \mathbf{r} , is independent of scale at which the image is formed.

IV. COMPARISON

In this section, the algorithms are compared with objective measures (mean square error) and subjective image quality. Image results from each of the algorithms are provided. For these experiments, simulated sampling was used where full-color images were sampled to match the CFA sampling process.

Twenty-four digital color images were used in the objective measure experiments. These images are part of the Kodak color image database and include various scenes. The images were sampled according to the Bayer CFA and reconstructed with a subset of the algorithms. Three measures were used to evaluate the algorithms. The mean square error was measured for each color plane on each of the output images to determine the difference between the original image and the reconstructed image. For a second measure, an extension of the CIELab metric was measured. The extension, ΔE_s , is described in [36] and a MATLAB code example is available online [1]. It measures error in a perceptually-uniform color space, extending the CIELab ΔE measure to account for non-uniform regions. A third measure used in

the evaluation is a measure of zipper effect [25]. Zipper effect is defined in the article as “an increase in color difference with respect to its most similar neighbor.” To determine if a pixel is affected by zipper effect, Lu and Tan compare the color change between neighboring pixels in the original, full-color image, and the demosaicked image. The original image is used to determine the most similar neighbor. If the color change exceeds a fixed threshold, that pixel is determined to have zipper effect. The error measure reports the percentage of pixels that have zipper effect.

The bar graph in Figure 9 shows the average MSE over the set of images, along with error bars showing the 25%-75% range for the set of images. The graph shows that the POCS method performs best on average in terms of MSE and the small range shown in the graph shows that it is also robust and performs well for all of the images.

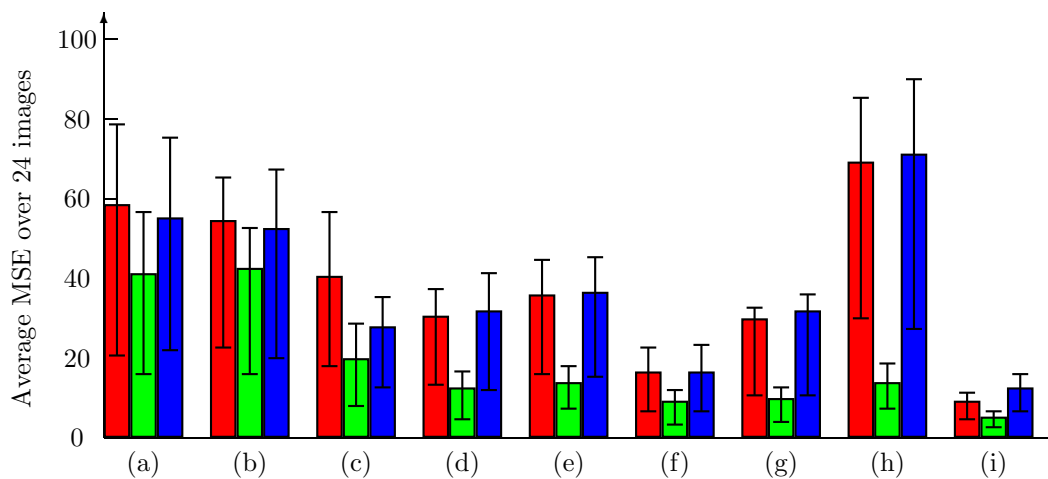


Fig. 9. Average mean square error for different algorithms. (a) Edge-directed interpolation in [17] (b) Constant hue-based interpolation in [3] (c) Weighted sum in [22] (d) Second-order gradients as correction terms in [4] (e) Bayesian approach in [26] (f) Homogeneity-directed in [18] (g) Pattern matching (Chang) in [8] (h) Alias cancellation in [12] (i) POCS in [14]

Table I reports the ΔE_s error and the percentage of pixels showing zipper effect. The errors are reported for the same set of algorithms. These measures agree with the mean square error comparison. The POCS method and the homogeneity-directed algorithm show superior results to the other algorithms.

The numbers can only provide part of the overall story. An important evaluation is the subjective appearance of the output images. For this, two example images are presented. An example of zipper effect artifacts can be seen in (d) and (e). Figure 10 shows the *Lighthouse* image. This example includes a picket fence from a perspective that increases spatial frequency along the fence. Aliasing is a prominent artifact in this image. The homogeneity-directed interpolation algorithm reconstructs this image best. Very

Algorithm	ΔE_s		Zipper Effect	
	Mean	IQR	Mean	IQR
(a)	2.1170	0.9444	0.2383	0.1519
(b)	1.6789	0.7599	0.2501	0.1628
(c)	2.0455	0.7647	0.0718	0.0208
(d)	1.4106	0.5539	0.2114	0.1291
(e)	1.3544	0.6980	0.2449	0.1716
(f)	0.9751	0.5960	0.0509	0.0406
(g)	1.3908	0.5644	0.1244	0.0692
(h)	1.6030	0.6681	0.4484	0.2186
(i)	0.9688	0.4619	0.0566	0.0488

TABLE I

A SPATIAL EXTENSION OF CIELAB AND A MEASURE OF ZIPPER EFFECT. THE ZIPPER EFFECT IS REPORTED IN PERCENTAGE OF PIXELS SHOWING ZIPPER EFFECT ACCORDING TO THE MEASURE. REPORTED IS THE AVERAGE ERROR OVER 24 IMAGES AND THE INTERQUARTILE RANGE. (A) EDGE-DIRECTED INTERPOLATION IN [17] (B) CONSTANT HUE-BASED INTERPOLATION IN [3] (C) WEIGHTED SUM IN [22] (D) SECOND-ORDER GRADIENTS AS CORRECTION TERMS IN [4] (E) BAYESIAN APPROACH IN [26] (F) HOMOGENEITY-DIRECTED IN [18] (G) PATTERN MATCHING (CHANG) IN [8] (H) ALIAS CANCELLATION IN [12] (I) POCS IN [14]

little aliasing is present in the output image. The *Boat* image in Figure 11 contains lines at various angles across the image. This is a good example to show how the algorithms respond to features at various orientations. The POCS algorithm and the homogeneity-directed interpolation algorithm show very few of the aliasing artifacts present in the other output images. This shows that these algorithms are fairly robust to the orientation of various features. According to the MSE measurements, POCS is the best algorithm, but the output images from the homogeneity-directed method have fewer artifacts. This suggests the need to use subjective evaluations along with objective measures.

In [24], Longere et al. provide a perceptual assessment of demosaicking algorithms. They compare several algorithms with a subjective experiment. The results of their first experiment show that the subjects favored sharpness and the algorithms providing a sharp image were highly favored. The experiment is repeated with the result images normalized for sharpness. After this adjustment, the results show more variation and no one algorithm is highly favored. Another comparison of demosaicking algorithms is provided in [29].

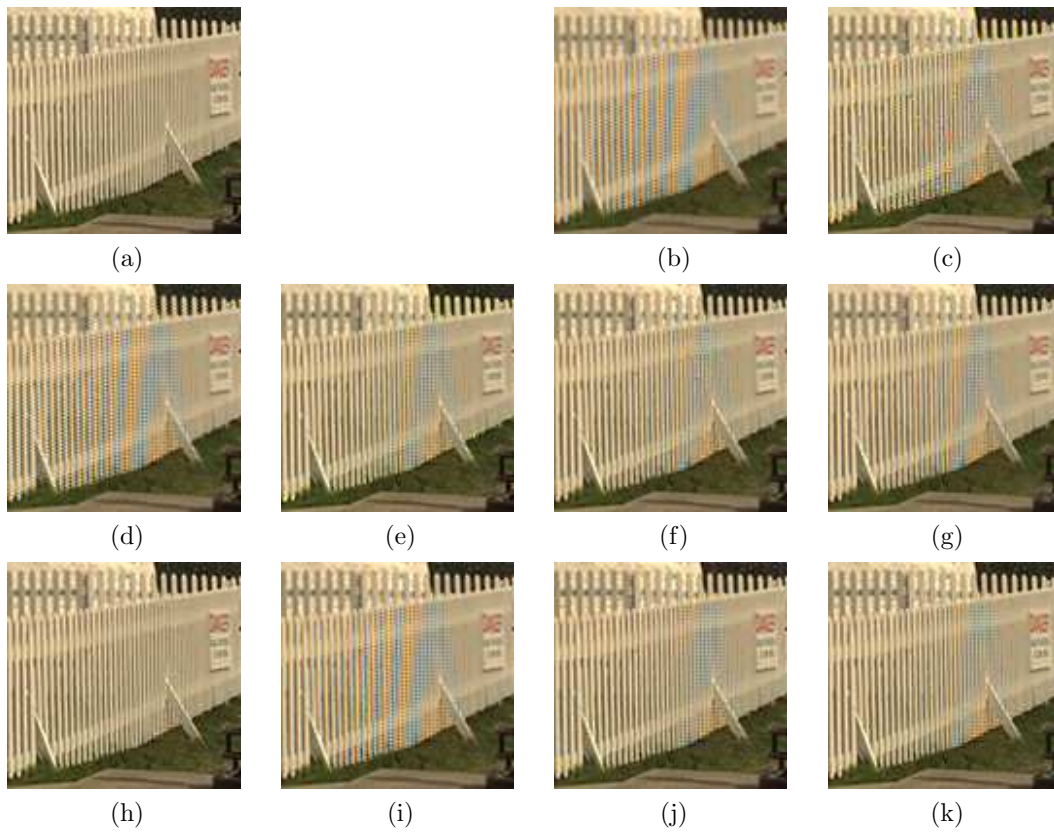


Fig. 10. Result images for example lighthouse image. (a) Original image (b) Bilinear interpolation (c) Edge-directed interpolation in [17] (d) Constant hue-based interpolation in [3] (e) Weighted sum in [22] (f) Second-order gradients as correction terms in [4] (g) Bayesian approach in [26] (h) Homogeneity-directed in [18] (i) Pattern matching (Chang) in [8] (j) Alias cancellation in [12] (k) POCS in [14]

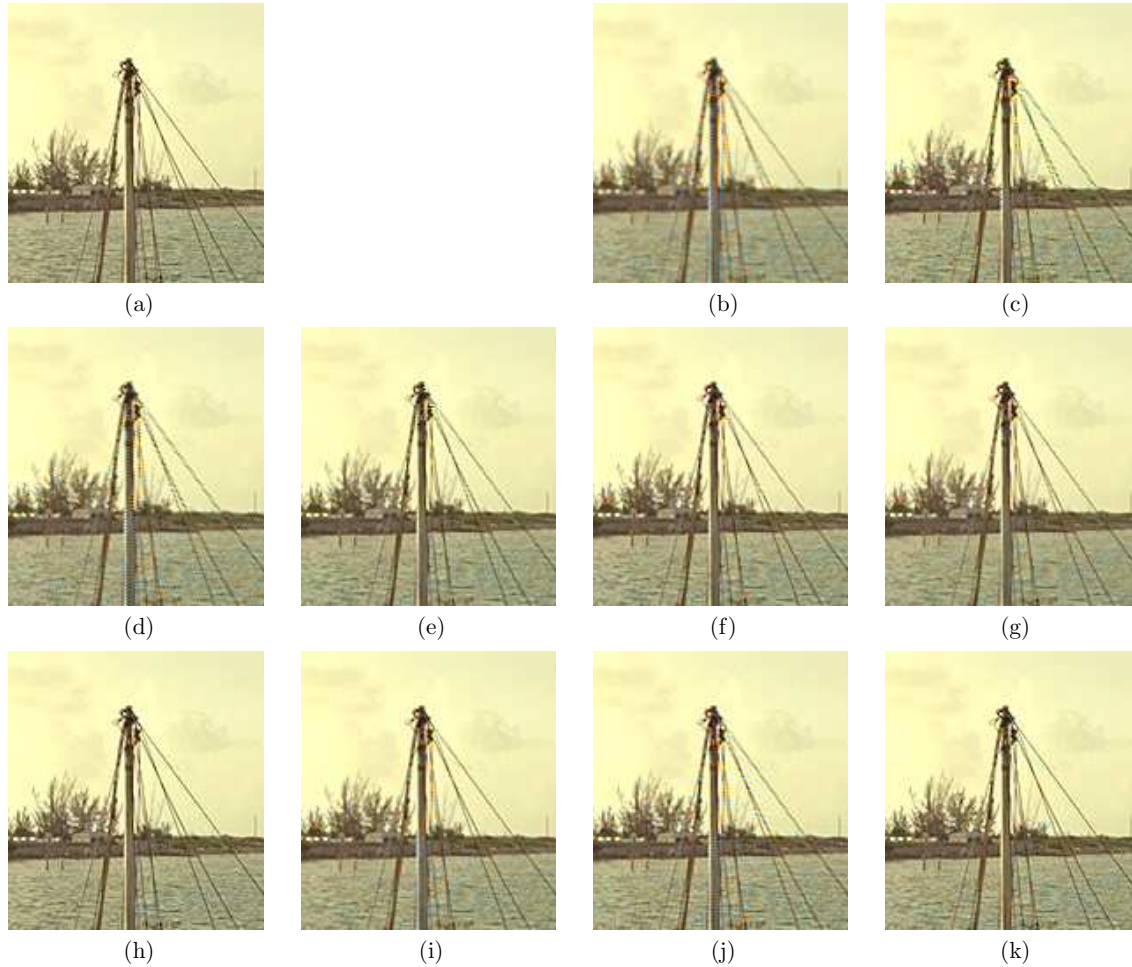


Fig. 11. Result images for example boat image. (a) Original image (b) Bilinear interpolation (c) Edge-directed interpolation in [17] (d) Constant hue-based interpolation in [3] (e) Weighted sum in [22] (f) Second-order gradients as correction terms in [4] (g) Bayesian approach in [26] (h) Homogeneity-directed in [18] (i) Pattern matching (Chang) in [8] (j) Alias cancellation in [12] (k) POCS in [14]

V. CONCLUSIONS AND FUTURE DIRECTIONS

The sensor size of digital cameras continues to decrease, providing sensor arrays with larger numbers of pixels. Today, five and six mega-pixel cameras are common. The increased sampling rate of these cameras reduces the probability of aliasing and other artifacts. Also, Foveon has invented an imaging sensor, the X3 sensor, that is able to capture red, green, and blue information at every pixel – eliminating the need for demosaicking in the digital camera pipeline.

However, research into the demosaicking problem is still an important problem. This research has provided an understanding of the image modelling process. The estimation methods discussed in this

article describe the image formation process, describing how the natural scene is transformed into a digital image. The correlation between the three color planes has also been explored. This extends beyond three color planes into hyperspectral image processing.

Processing time is often an important measure for algorithms implemented in real-time systems. A photographer needs to be able to take pictures at a fast rate and the image processing can sometimes limit this. Several cameras, especially the more expensive digital SLR cameras, provide access to the raw image data captured by the sensor. With this data, the images can be processed at a later time on a computer. In this case, processing time is not critically important. Therefore, algorithms that perform well, but are computationally complex, can still be considered in off-line processing applications.

REFERENCES

- [1] [Online]. Available: <http://white.stanford.edu/~brian/scielab/scielab.html>
- [2] J. Adams, K. Parulski, and K. Spaulding, "Color processing in digital cameras," *IEEE Micro*, vol. 18, no. 6, pp. 20–31, 1998.
- [3] J. E. Adams, "Interactions between color plane interpolation and other image processing functions in electronic photography," in *Proc. SPIE*, vol. 2416, 1995, pp. 144–151.
- [4] J. E. Adams and J. F. Hamilton, "Design of practical color filter array interpolation algorithms for digital cameras," in *Proc. SPIE*, vol. 3028, 1997, pp. 117–125.
- [5] D. Alleysson, S. Susstrunk, and J. Herault, "Color demosaicing by estimating luminance and opponent chromatic signals in the fourier domain," in *Proc. Color Imaging Conference: Color Science, Systems, and Applications*, 2002, pp. 331–336.
- [6] Z. Baharav and R. Kakarala, "Compression aware demosaicing methods," in *Proc. SPIE*, vol. 4667, 2002, pp. 149–156.
- [7] D. H. Brainard, "Bayesian method for reconstructing color images from trichromatic samples," in *Proc. IS & T 47th Annual Meeting*, 1994, pp. 375–380.
- [8] E. Chang, S. Cheung, and D. Pan, "Color filter array recovery using a threshold-based variable number of gradients," in *Proc. SPIE*, vol. 3650, 1999, pp. 36–43.
- [9] D. R. Cok, "Signal processing method and apparatus for producing interpolated chrominance values in a sampled color image signal," *U.S. Patent 4,642,678*, 1986.
- [10] W. T. Freeman, "Method and apparatus for reconstructing missing color samples," *U.S. Patent 4,774,565*, 1988.
- [11] S. Geman and D. Geman, "Stochastic relaxation, gibbs distributions and the bayesian distribution of images," *IEEE Trans. Pattern Analysis and Machine Intelligence*, no. 6, pp. 721–741, 1984.
- [12] 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 Processing*, vol. 1, 2001, pp. 141–144.
- [13] J. Go, K. Sohn, and C. Lee, "Interpolation using neural networks for digital still cameras," *IEEE Trans. Consumer Electronics*, vol. 46, no. 3, pp. 610–616, August 2000.
- [14] B. K. Gunturk, Y. Altunbasak, and R. M. Mersereau, "Color plane interpolation using alternating projections," *IEEE Trans. Image Processing*, vol. 11, no. 9, pp. 997–1013, September 2002.
- [15] M. R. Gupta and T. Chen, "Vector color filter array interpolation," in *Proc. SPIE*, vol. 4306, 2001, pp. 374–382.

- [16] Y. Hel-Or and D. Keren, "Image demosaicing method utilizing directional smoothing," *U.S. Patent 6,404,918*, 2002.
- [17] R. H. Hibbard, "Apparatus and method for adaptively interpolating a full color image utilizing luminance gradients," *U.S. Patent 5,382,976*, 1995.
- [18] K. Hirakawa and T. W. Parks, "Adaptive homogeneity-directed demosaicing algorithm," *Proc. IEEE Int. Conf. Image Processing*, vol. 3, pp. 669–672, 2003.
- [19] R. Kakarala and Z. Baharav, "Adaptive demosaicing with the principal vector method," *IEEE Trans. Consumer Electronics*, vol. 48, no. 4, pp. 932–937, November 2002.
- [20] O. Kapah and H. Z. Hel-Or, "Demosaicing using artificial neural networks," in *Proc. SPIE*, vol. 3962, 2000, pp. 112–120.
- [21] D. Keren and M. Osadchy, "Restoring subsampled color images," *Machine Vision and Applications*, vol. 11, no. 4, pp. 197–202, December 1999.
- [22] R. Kimmel, "Demosaicing: image reconstruction from ccd samples," *IEEE Trans. Image Processing*, vol. 8, pp. 1221–1228, 1999.
- [23] C. A. Laroche and M. A. Prescott, "Apparatus and method for adaptively interpolating a full color image utilizing chrominance gradients," *U.S. Patent 5,373,322*, 1994.
- [24] P. Longere, X. Zhang, P. B. Delahunt, and D. H. Brainard, "Perceptual assessment of demosaicing algorithm performance," *Proc. IEEE*, vol. 90, no. 1, pp. 123–132, January 2002.
- [25] W. Lu and Y.-P. Tan, "Color filter array demosaicking: New method and performance measures," *IEEE Trans. Image Processing*, vol. 12, no. 10, pp. 1194–1210, October 2003.
- [26] J. Mukherjee, R. Parthasarathi, and S. Goyal, "Markov random field processing for color demosaicing," *Pattern Recognition Letters*, vol. 22, no. 3-4, pp. 339–351, March 2001.
- [27] S.-C. Pei and I.-K. Tam, "Effective color interpolation in ccd color filter arrays using signal correlation," *IEEE Trans. Circuits and Systems for Video Technology*, vol. 13, no. 6, pp. 503–513, June 2003.
- [28] R. Ramanath and W. E. Snyder, "Adaptive demosaicking," *Journal of Electronic Imaging*, vol. 12, no. 4, pp. 633–642, October 2003.
- [29] R. Ramanath, W. E. Snyder, G. L. Bilbro, and W. A. S. III, "Demosaicking methods for bayer color arrays," *Journal of Electronic Imaging*, vol. 11, no. 3, pp. 306–315, July 2002.
- [30] B. Tao, I. Tastl, T. Cooper, M. Blasgen, and E. Edwards, "Demosaicing using human visual properties and wavelet interpolation filtering," in *Proc. Color Imaging Conference: Color Science, Systems, and Applications*, 1999, pp. 252–256.
- [31] D. Taubman, "Generalized wiener reconstruction of images from colour sensor data using a scale invariant prior," in *Proc. IEEE Int. Conf. Image Processing*, vol. 3, 2000, pp. 801–804.
- [32] H. J. Trussell and R. E. Hartwig, "Mathematics for demosaicking," *IEEE Trans. Image Processing*, vol. 3, no. 11, pp. 485–492, April 2002.
- [33] P.-S. Tsai, T. Acharya, and A. K. Ray, "Adaptive fuzzy color interpolation," *Journal of Electronic Imaging*, vol. 11, no. 3, pp. 293–305, July 2002.
- [34] J. A. Weldy, "Optimized design for a single-sensor color electronic camera system," *Proc. SPIE*, vol. 1071, pp. 300–307, 1988.
- [35] X. Wu, W. K. Choi, and P. Bao, "Color restoration from digital camera data by pattern matching," *Proc. SPIE*, vol. 3018, pp. 12–17, 1997.
- [36] X. Zhang and B. A. Wandell, "A spatial extension of cielab for digital color image reproduction," in *Society for Information Display Symposium Technical Digest*, vol. 27, 1996, pp. 731–734.