

Demosaicking methods for Bayer color arrays

Rajeev Ramanath

Wesley E. Snyder

Griff L. Bilbro

North Carolina State University

Department of Electrical and Computer Engineering

Raleigh, North Carolina 27695-7914

E-mail: rramana@eos.ncsu.edu

William A. Sander III

U.S. Army Research Office

Durham

P.O. Box 12211

Research Triangle Park, North Carolina 27709

Abstract. Digital Still Color Cameras sample the color spectrum using a monolithic array of color filters overlaid on a charge coupled device array such that each pixel samples only one color band. The resulting mosaic of color samples is processed to produce a high resolution color image such that the values of the color bands not sampled at a certain location are estimated from its neighbors. This process is often referred to as demosaicking. This paper introduces and compares a few commonly used demosaicking methods using error metrics like mean squared error in the RGB color space and perceived error in the CIELAB color space. © 2002 SPIE and IS&T. [DOI: 10.1117/1.1484495]

1 Introduction

Commercially available Digital Still Color Cameras are based on a single charge coupled device (CCD) array and capture color information by using three or more color filters, each sample point capturing only one sample of the color spectrum.

The Bayer array¹ [shown in Fig. 1(a)] is one of the many realizations of color filter arrays (CFA) possible. Many other implementations of a color-sampling grid have been incorporated in commercial cameras, most using the principle that the luminance channel (green) needs to be sampled at a higher rate than the chrominance channels (red and blue). The choice for green as “representative” of the luminance is due to the fact that the luminance response curve of the eye peaks at around the frequency of green light (around 550 nm).

Since, at each pixel, only one spectral measurement was made, the other colors must be estimated using information from all the color planes in order to obtain a high resolution color image. This process is often referred to as *demosaicking*. Interpolation must be performed on the mosaicked image data. There are a variety of methods available, the sim-

plest being linear interpolation, which, as shall be shown, **does not maintain edge information well**. More complicated methods^{2–6} perform this interpolation and attempt to maintain edge detail or limit hue transitions. In Ref. 7, Trussell introduces a linear lexicographic model for the image formation and demosaicking process, which may be used in a reconstruction step. In Ref. 8, linear response models proposed by Vora *et al.*⁹ have been used to reconstruct these mosaicked images using an optimization technique called **mean field annealing**.¹⁰ In this paper we briefly describe the more commonly used demosaicking algorithms and demonstrate their strengths and weaknesses. In Sec. 2, we describe the interpolation methods we use in our comparisons. We compare the interpolation methods by running the algorithms on three types of images (two types of synthetic image sets and one set of real-world mosaicked images). The images used for comparison and their properties are presented in Sec. 3. Qualitative and quantitative results are presented in Sec. 4. Discussions about the properties of these algorithms and their overall behavior are presented in Sec. 5. We use two error metrics, **the mean squared error in the RGB color space and the ΔE_{ab}^* error in the CIELAB color space** (described in the Appendix).

2 Demosaicking Strategies

2.1 Ideal Interpolation

Sampling of a continuous image $f(x,y)$ yields infinite repetitions of its continuous spectrum $F(\zeta,\eta)$ in the Fourier domain. If these repetitions do not overlap (which is almost never the case as natural images are not band limited), the original image $f(x,y)$ can be reconstructed exactly from its discrete samples $f(m,n)$, otherwise we observe the phenomenon of **aliasing**. The one-dimensional “ideal” interpolation is the multiplication with a rect function in the frequency domain and can be realized in the spatial domain by a convolution with the sinc function. This “ideal interpola-

| | | | | | | |
|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|
| R ₁₁ | G ₁₂ | R ₁₃ | G ₁₄ | R ₁₅ | G ₁₆ | R ₁₇ |
| G ₂₁ | B ₂₂ | G ₂₃ | B ₂₄ | G ₂₅ | B ₂₆ | G ₂₇ |
| R ₃₁ | G ₃₂ | R ₃₃ | G ₃₄ | R ₃₅ | G ₃₆ | R ₃₇ |
| G ₄₁ | B ₄₂ | G ₄₃ | R ₄₄ | G ₄₅ | B ₄₆ | G ₄₇ |
| R ₅₁ | G ₅₂ | R ₅₃ | G ₅₄ | R ₅₅ | G ₅₆ | R ₅₇ |
| G ₆₁ | B ₆₂ | G ₆₃ | B ₆₄ | G ₆₅ | B ₆₆ | G ₆₇ |
| R ₇₁ | G ₇₂ | R ₇₃ | G ₇₄ | R ₇₅ | G ₇₆ | R ₇₇ |

Fig. 1 Sample Bayer pattern.

tor” kernel is band limited and, hence, is not space limited. It is primarily of theoretical interest and not implemented in practice.¹¹

2.2 Neighborhood Considerations

It may be expected that we get better estimates for the missing sample values by increasing the neighborhood of the pixel, but this increase is computationally expensive. There is, hence, a need to keep the interpolation filter kernel space-limited to a small size and also extract as much information from the neighborhood as possible. To this end, correlation between color channels is used.¹² For RGB images, crosscorrelation between channels has been determined and found to vary between 0.25 and 0.99 with averages of 0.86 for red/green, 0.79 for red/blue, and 0.92 for green/blue cross correlations.¹³ One well-known image model¹² is to simply assume that red and blue are perfectly correlated with the green over a small neighborhood and thus differ from green by only an offset. This image model is given by

$$G_{ij} = R_{ij} + k, \quad (1)$$

where (i, j) refers to the pixel location, R (known) and G (unknown) the red and green pixel values, k is the appropriate bias for the given pixel neighborhood. The same applies at a blue pixel location. The choice of the neighborhood size in such a case is important. It is observed that most implementations are designed with hardware imple-

mentation in mind (paying great attention to the need for pipelining, system latency, and throughput per clock cycle). The larger the neighborhood, the greater the difficulty in pipelining, the greater the latency, and possibly, lesser the throughput.

2.3 Bilinear Interpolation

Consider the array of pixels as shown in Fig. 1(a). At a blue center (where blue color was measured), we need to estimate the green and red components. Consider pixel location 44 at which only B_{44} is measured; we need to estimate G_{44} . Given G_{34} , G_{43} , G_{45} , G_{54} , one estimate for G_{44} is given by $G_{44} = (G_{34} + G_{43} + G_{45} + G_{54})/4$. To determine R_{44} , given R_{33} , R_{35} , R_{53} , R_{55} , the estimate for R_{44} is given by $R_{44} = (R_{33} + R_{35} + R_{53} + R_{55})/4$. At a red center, we would estimate the blue and green accordingly. Performing this process at each photosite (location on the CCD), we can obtain three color planes for the scene which would give us one possible demosaicked form of the scene.

The band-limiting nature of this interpolation smooths edges, which shows up in color images as fringes (referred to as the zipper effect.^{12,14}) This has been illustrated with two colors channels (for simplicity) in Fig. 2.

2.4 Constant Hue-Based Interpolation

In general, hue is defined as the property of colors by which they can be perceived as ranging from red through yellow, green, and blue, as determined by the dominant wavelength of the light. Constant hue-based interpolation, proposed by Cok² and is one of the first few methods used in commercial camera systems. Modifications of this system are still in use. The key objection to pixel artifacts in images that result from bilinear interpolation is abrupt and unnatural hue change.² There is a need to maintain the hue of the color such that there are no sudden jumps in hue (except for over edges, say). The red and blue channels are assigned to be the chrominance channels while the green channel is assigned as the luminance channel.

As used in this section, hue is defined by a vector of ratios as $(R/G, B/G)$.² It is to be noted that the term hue defined above is valid for this method only, also, the hue needs to be “redefined” if the denominator G is zero. By interpolating the hue value and deriving the interpolated chrominance values (blue and red) from the interpolated hue values, hues are allowed to change only gradually, thereby reducing the appearance of color fringes which would have been obtained by interpolating only the chrominance values.

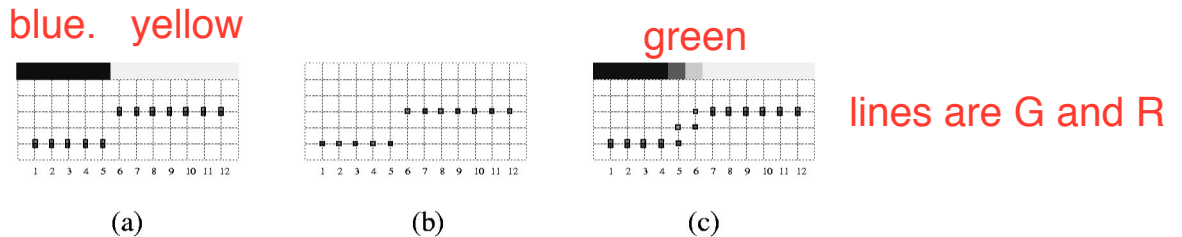


Fig. 2 Illustration of fringe or zipper effect resulting from the linear interpolation process. An edge is illustrated as going from navy blue (0,0,128) to yellow (255,255,128). The zipper effect produces green pixels near the edge: (a) original image (only 2 colors, blue constant at 128), (b) one scan line of subsampled Bayer pattern (choose every other pixel), (c) result of estimating missing data using linear interpolation. Observe color fringe in locations 5 and 6.

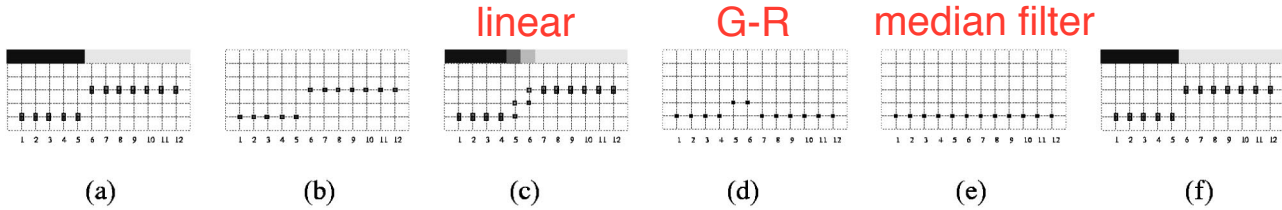


Fig. 3 Illustration of Freeman's interpolation method for a two channel system, as in Fig. 2 an edge is illustrated as going from navy blue (0,0,128) to yellow (255,255,128): (a) original image (only 2 colors, blue constant at 128), (b) one scan line of subsampled Bayer pattern (choose every other pixel), (c) result of linear interpolation, (d) green minus red, (e) median filtered result (filter size of **five** pixels) of the difference image, and (f) reconstructed image.

Consider an image with constant hue. In **exposure** space {be it logarithmic [Most cameras capture data in a logarithmic exposure space and need to be linearized before the ratios used as such. If interpolating in the logarithmic exposure space, difference of logarithms needs to be taken instead of ratios; i.e., $\log(R_{ij}/R_{kl}) = \log(R_{ij}) - \log(R_{kl})$] or linear}, the values of the luminance (G) and one chrominance component (R , say) at a location (i,j) and a neighboring sample location (k,l) are related as $R_{ij}/R_{kl} = G_{ij}/G_{kl}$ if $B_{ij}/B_{kl} = G_{ij}/G_{kl}$.

If R_{kl} represents the unknown chrominance value, and R_{ij} and G_{ij} represent measured values and G_{kl} represents the **interpolated luminance value**, the missing chrominance value R_{kl} is given by $R_{kl} = G_{kl}(R_{ij}/G_{ij})$. In an image that does not have uniform hue, as in a typical color image, smoothly changing hues are assured by **interpolating the hue values between neighboring chrominance values**.

The green channel is first interpolated using bilinear interpolation. After this first pass, the hue is interpolated. Referring to Fig. 2(a),

$$R_{44} = G_{44} \frac{\frac{R_{33}}{G_{33}} + \frac{R_{35}}{G_{35}} + \frac{R_{53}}{G_{53}} + \frac{R_{55}}{G_{55}}}{4} \quad (2)$$

and similarly for the blue channel

$$B_{33} = G_{33} \frac{\frac{B_{22}}{G_{22}} + \frac{B_{24}}{G_{24}} + \frac{B_{42}}{G_{42}} + \frac{B_{44}}{G_{44}}}{4}. \quad (3)$$

The G values in boldface are estimated values, after the first pass of interpolation. **The extension to the logarithmic exposure space is straightforward as multiplications and divisions in the linear space become additions and subtractions, respectively, in the logarithmic space.** There is a caveat however as interpolations will be performed in the logarithmic space and, hence, the relations in linear space and exposure space are not identical.² Hence in most implementations the data is first linearized¹⁵ and then interpolated as described earlier.

2.5 Median-Based Interpolation

This method, proposed by **Freeman**,³ is a two pass process, the first being a linear interpolation, and the second pass a median filter of the color differences.

In the first pass, linear interpolation is used to populate each photosite with all three colors and in the second pass,

the difference image, of say, **red minus green and blue minus green is median filtered**. The median filtered image thus obtained is then used in conjunction with the original Bayer array samples to recover the samples (illustrated below). This method preserves edges well, as illustrated in Fig. 3 where only one row of the Bayer array is considered since this process can be extrapolated to the case of the rows containing blue and green pixels. Figure 3(a) shows one scan line of the original image before Bayer subsampling, the horizontal axis is the location index and the **vertical axis represents intensity of red and green pixels**. We have a step edge between locations 5 and 6. Figure 3(b) shows the same scan line, sampled in a Bayer fashion, picking out every other pixel for red and green. Figure 3(c) (step 1 of this algorithm) shows the result of estimating the missing data using **linear interpolation**. Notice the color fringes introduced between pixel locations 5 and 6; Fig. 3(d) (step 2) shows the **absolute valued difference image between the two channels**; Fig. 3(e) (step 3) shows the result of median filtering the difference image with a kernel of size 5. Using this result and the sampled data, Fig. 3(f) is generated (step 4) as an estimate of the original image **(by adding the median filtered result to the sampled data, e.g., the red value at location 6 is estimated by adding the median filtered result at location 6 to the sampled green value at location 6)**. The reconstruction of the edge in this example is exact, although note that for a median filter of size 3, this will not be the case.

This concept can be carried over to three color sensors wherein differences are calculated between pairs of colors and the median filter is applied to these differences to generate the final image.

We shall consider neighborhoods of a size such that all the algorithms can be compared on the same basis. The algorithms described in this document have at most nine pixels under consideration for "estimation." In a square neighborhood, this would imply a 3×3 window. We shall, hence, use a 3×3 neighborhood for Freeman's algorithm.

2.6 Gradient Based Interpolation

This method was proposed by Laroche and Prescott⁴ and is in use in the Kodak DCS 200 digital camera system. It employs a three step process, the first one being the interpolation of the luminance channel (green) and the second and third being interpolation of the color differences (red minus green and blue minus green). The interpolated color differences are used to reconstruct the chrominance channels (red and blue). This method takes advantage of the fact that the human eye is most sensitive to luminance changes.

The interpolation is performed depending upon the position of an edge in the green channel. Referring to Fig. 3(a), if we need to estimate G_{44} , let $\alpha = \text{abs}[(B_{42} + B_{46})/2 - B_{44}]$ and $\beta = \text{abs}[(B_{24} + B_{64})/2 - B_{44}]$. We refer to α and β as “classifiers” and will use them to determine if a pixel belongs to a vertical or horizontal edge, respectively. It is intriguing to note that the classifiers used are second derivatives with the sign inverted and halved in magnitude. We come up with the following estimates for the missing green pixel value:

$$\mathbf{G}_{44} = \begin{cases} \frac{G_{43} + G_{45}}{2} & \text{if } \alpha < \beta \\ \frac{G_{34} + G_{54}}{2} & \text{if } \alpha > \beta \\ \frac{G_{43} + G_{45} + G_{34} + G_{54}}{4} & \text{if } \alpha = \beta \end{cases} \quad (4)$$

Similarly, for estimating G_{33} , let $\alpha = \text{abs}[(R_{31} + R_{35})/2 - R_{33}]$ and $\beta = \text{abs}[(R_{13} + R_{53})/2 - R_{33}]$. These are estimates to the horizontal and vertical second derivatives in red, respectively. Using these gradients as classifiers, we come up with the following estimates for the missing green pixel value:

$$\mathbf{G}_{33} = \begin{cases} \frac{G_{32} + G_{34}}{2} & \text{if } \alpha < \beta \\ \frac{G_{23} + G_{43}}{2} & \text{if } \alpha > \beta \\ \frac{G_{32} + G_{34} + G_{23} + G_{43}}{4} & \text{if } \alpha = \beta \end{cases} \quad (5)$$

Once the luminance is determined, the chrominance values are interpolated from the differences between the color (red and blue) and luminance (green) signals. This is given by

$$\begin{aligned} R_{34} &= \frac{(R_{33} - \mathbf{G}_{33}) + (R_{35} - \mathbf{G}_{35})}{2} + G_{34}, \\ R_{43} &= \frac{(R_{33} - \mathbf{G}_{33}) + (R_{35} - \mathbf{G}_{35})}{2} + G_{43}, \\ R_{44} &= \frac{(R_{33} - \mathbf{G}_{33}) + (R_{35} - \mathbf{G}_{35}) + (R_{53} - \mathbf{G}_{53}) + (R_{55} - \mathbf{G}_{55})}{4} \\ &\quad + G_{44}. \end{aligned} \quad (6)$$

Note that the green channel has been completely estimated before this step. The boldface entries correspond to estimated values. We get corresponding formulas for the blue pixel locations. Interpolating color differences and adding the green component has the advantage of maintaining color information and also using intensity information at pixel locations. At this point, three complete RGB planes are available for the full resolution color image.

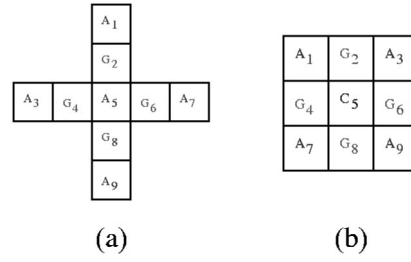


Fig. 4 Sample Bayer neighborhood, A_i =chrominance (blue/red), G_i =luminance, C_5 =red/blue.

2.7 Adaptive Color Plane Interpolation

This method is proposed by Hamilton and Adams.⁵ It is a modification of the method proposed by Laroche and Prescott.⁴ This method also employs a multiple step process, with classifiers similar to those used in Laroche–Prescott’s scheme but modified to accommodate first order and second order derivatives. The estimates are composed of arithmetic averages for the chromaticity (red and blue) data and appropriately scaled second derivative terms for the luminance (green) data. Depending upon the preferred orientation of the edge, the predictor is chosen. This process also has three runs. The first run populates that luminance (green) channel and the second and third runs populate the chrominance (red and blue) channels.

Consider the Bayer array neighborhood shown in Fig. 4(a). G_i is a green pixel and A_i is either a red pixel or a blue pixel (all A_i pixels will be the same color for the entire neighborhood). We now form classifiers $\alpha = \text{abs}(-\mathbf{A}_3 + 2\mathbf{A}_5 - \mathbf{A}_7) + \text{abs}(G_4 - G_6)$ and $\beta = \text{abs}(-\mathbf{A}_1 + 2\mathbf{A}_5 - \mathbf{A}_9) + \text{abs}(G_2 - G_8)$. These classifiers are composed of second derivative terms for chromaticity data and gradients for the luminance data. As such, these classifiers sense the high spatial frequency information in the pixel neighborhood in the horizontal and vertical directions.

Consider, that we need to estimate the green value at the center, i.e., to estimate G_5 . Depending upon the preferred orientation, the interpolation estimates are determined as

$$\mathbf{G}_5 = \begin{cases} \frac{G_4 + G_6}{2} + \frac{-\mathbf{A}_3 + 2\mathbf{A}_5 - \mathbf{A}_7}{2} & \text{if } \alpha < \beta \\ \frac{G_2 + G_8}{2} + \frac{-\mathbf{A}_1 + 2\mathbf{A}_5 - \mathbf{A}_9}{2} & \text{if } \alpha > \beta \\ \frac{G_2 + G_4 + G_6 + G_8}{4} + \frac{-\mathbf{A}_1 - \mathbf{A}_3 + 4\mathbf{A}_5 - \mathbf{A}_7 - \mathbf{A}_9}{8} & \text{if } \alpha = \beta \end{cases} \quad (7)$$

These predictors are composed of arithmetic averages for the green data and appropriately scaled second derivative terms for the chromaticity data. This comprises the first pass of the interpolation algorithm. The second pass involves populating the chromaticity channels. Consider the neighborhood as shown in Fig. 4(b). G_i is a green pixel, A_i is either a red pixel or a blue pixel and C_i is the opposite chromaticity pixel. Then $A_2 = (A_1 + A_3)/2 + (-\mathbf{G}_1 + 2\mathbf{G}_2 - \mathbf{G}_3)/2$, $A_4 = (A_1 + A_7)/2 + (-\mathbf{G}_1 + 2\mathbf{G}_4 - \mathbf{G}_7)/2$. These are used when the nearest neighbors to A_i are in the same row and column respectively.

To estimate C_5 , we employ the same method as we did to estimate the luminance channel. We again, form two classifiers, α and β which “estimate” the gradient in the horizontal and vertical directions. $\alpha = \text{abs}(-\mathbf{G}_3 + 2\mathbf{G}_5 - \mathbf{G}_7) + \text{abs}(A_3 - A_7)$ and $\beta = \text{abs}(-\mathbf{G}_1 + 2\mathbf{G}_5 - \mathbf{G}_9) + \text{abs}(A_1 - A_9)$. α and β “sense” the high frequency information in the pixel neighborhood in the positive and negative diagonal respectively. We now have estimates

$$C_5 = \begin{cases} \frac{A_3 + A_7}{2} + \frac{-\mathbf{G}_3 + 2\mathbf{G}_5 - \mathbf{G}_7}{2} & \text{if } \alpha < \beta \\ \frac{A_1 + A_9}{2} + \frac{-\mathbf{G}_1 + 2\mathbf{G}_5 - \mathbf{G}_9}{2} & \text{if } \alpha > \beta \\ \frac{A_1 + A_3 + A_7 + A_9}{4} + \frac{-\mathbf{G}_1 - \mathbf{G}_3 + 4\mathbf{G}_5 - \mathbf{G}_7 - \mathbf{G}_9}{4} & \text{if } \alpha = \beta \end{cases} \quad (8)$$

These estimates are composed of arithmetic averages for the chromaticity data and appropriately scaled second derivative terms for the green data. Depending upon the preferred orientation of the edge, the predictor is chosen. We now have the three color planes populated for the Bayer array data.

3 Comparison of Interpolation Methods

We generated test images, shown in Figs. 5 and 6 which are simulations of the data contained in the Bayer array of the camera. In other words, these are images that consider “what-if” cases in the Bayer array. They were chosen as test images to emphasize the various details that each algorithm works on.

3.1 Type I Test Images

Images of this type are synthetic and have edge orientations along both the cardinal directions as well as in arbitrary directions as shown in Fig. 5. Test image₁ was chosen to demonstrate the artifacts each process introduces for varying thicknesses of stripes (increasing spatial frequencies). Test image₂ was chosen to study a similar performance, but with a constant spatial frequency. Test image₃ is a section from the starburst pattern, to test the robustness of these algorithms for noncardinal edge orientations. Note that these images have perfectly correlated color planes. The intent of these images is to highlight alias-induced fringing errors.

3.2 Type II Images

Three RGB images, shown in Fig. 6 were subsampled in the form of a Bayer array and then interpolated to get the three color planes. The regions of interest (ROIs) in this image has been highlighted with a white box. These images were chosen specifically to highlight the behavior of these algorithms when presented with color edges. Test image₄ is a synthetic image of randomly chosen color patches. Unlike type I images, these images have sharp discontinuities in all color planes, independent of each other. The ROIs in Fig. 6(b) have relatively high spatial frequencies. The ROIs in Fig. 6(c) have distinct color edges, one between pastel colors and the other between fully saturated colors.

3.3 Type III Images

This category of images consists of real-world camera images captured with a camera that has a CFA pattern. No internal interpolation is performed on them. We were therefore able to get the “true” CFA imagery corrupted only by the optical PSF. The ROIs of these images are shown in Figs. 15(a) and 16(a). CFA₁ has sharp edges and high frequency components while CFA₂ has a color edge.

4 Results

The results of the demosaicking algorithms presented in Sec. 2 on the three types of images are shown in Figs. 7–16. Literature¹⁶ suggests that the ΔE_{ab}^* (definition included in the Appendix) error metric represents human perception effectively. We, hence, make use of this to quantify the errors observed. However, bear in mind the bounds on this error for detectability that ΔE_{ab}^* errors less than about 2.3 are not easily detected while on the other hand, errors greater than about 10 are so large that relative comparison is insignificant.¹⁷ This metric gives us a measure of the difference between colors as viewed by a standard observer. Another metric used for comparison is the mean squared error (MSE) which provides differences between colors in a “Euclidean” sense. MSE, although not being representative of the errors we perceive, is popular because of its tractability and ease in implementation. These metrics are tabulated in Tables 1 and 2. The boldface numbers represent the minimum values in the corresponding image, which gives us an idea about which algorithm performs best for a given image. There will be errors introduced in the printing/reproduction process, but assuming that the errors will be consistent for all the reproductions, we may infer relative performance of these algorithms.

In Figs. 7 and 8, notice the fringe artifacts introduced in linear interpolation, termed as the zipper effect by Adams.¹² The appearance of this effect is considerably reduced (observe the decrease in the metrics) in Cok’s interpolation. Hamilton–Adams’ and Laroche–Prescott’s implementation estimates test image₂ exactly (notice that the MSE and ΔE_{ab}^* errors are zero). This is because both these algorithms use information from the other channels for estimation (chrominance channel to interpolate luminance and vice versa). Notice that all these algorithms perform poorly at high spatial frequencies. All the algorithms discussed here have identical properties in the horizontal and vertical directions.

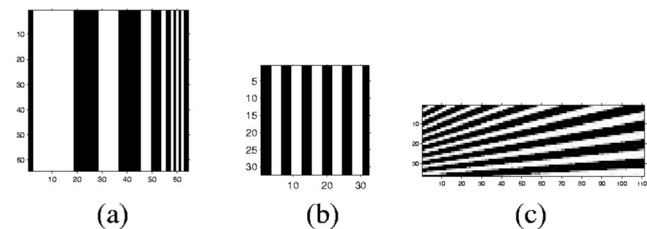


Fig. 5 Type I test images: (a) Test image₁ has vertical bars with decreasing thicknesses (16 pixels down to 1 pixel), (b) test image₂ has bars of constant width (3 pixels), and (c) test image₃ is a section from the starburst pattern.

For noncardinal edge orientations such as those shown in test image₃ (Fig. 9) performance (observed in the error metrics also) is noted to be worse. Note that the ΔE_{ab}^* error metric is “on an average” considerably higher for test image₃ when compared to test image₁ and test image₂.

Test image₄ has been used to illustrate the performance of these algorithms when presented with sharp edges which do not have correlated color planes (see Fig. 10). From the error metrics, it is clear that all of them perform poorly at sharp color edges. Note however that although the ΔE_{ab}^* errors are high, the squared error metric is relatively low, clearly highlighting the advantage of using ΔE_{ab}^* . Using only the squared error would have been misleading.

The macaw images illustrate the alias-induced errors while at the same time, showing a confetti type of error. These errors come about due to intensely bright or dark points (in a dark or bright neighborhood, respectively). Freeman’s algorithm performs best in these regions because it is able to remove such “speckle” behavior in the images due to the median filtering process (observe that the ΔE_{ab}^* errors are smallest for Freeman’s algorithms in such regions).

The crayon images on the other hand are reproduced precisely (see Figs. 13 and 14), with few errors. ROI₁ shows some errors at the edges where the line-art appears. However, this error is not evident. ROI₂ is reproduced almost exactly. In fact, depending upon the print process or the display rendering process, one may not be able to see the errors generated at all. This shows that these algorithms perform well at blurred color edges (which is the case with many natural scenes).

In type III images which are raw readouts from a CFA camera, we cannot use the metrics we have been using thus far as there is no “reference” image with which to compare these results. However, we may use visual cues to determine performance, and we observe similar trends in these images as was observed in synthetic images. Observe in Fig. 15 that the high spatial frequencies and noncardinal edge orientations are not reproduced correctly (as was the case with type I images). Color edges are also reproduced with reasonably good fidelity as is seen in Fig. 16—although some zipper effect is observed with Linear and Cok interpolations.

5 Discussion

Laroche–Prescott’s and Hamilton–Adams’ interpolation processes have similar forms. Both of them use second derivatives to perform interpolation which may be written as

$$v = u + \lambda g, \quad (9)$$

where u is the data (original image), v is the resulting image $\lambda > 0$, and g is a suitably defined gradient. We may think of Eq. (9) in the form of that used for unsharp masking,¹⁸ an enhancement process. Unsharp masking may be interpreted as either subtraction of the low-pass image from the original image (scaled) or of even as addition of a high-pass image to the original image (scaled). To see the equivalence let the image I be written as

$$I = L + H \quad (10)$$

the sum of its low-pass (L) and high-pass (H) components. Now, define unsharp masking by

$$F = aI - L = (a - 1)I + I - L = (a - 1)I + H, \quad (11)$$

which has a form similar to that in Eq. (9). Hence, one of the many ways to interpret Laroche–Prescott’s and Hamilton–Adams’ algorithms, is an unsharp masking process. It may, hence, be expected that these processes will sharpen edges (only those in the cardinal directions, due to the manner in which they are implemented) in the resulting images as is observed in the results obtained from Laroche–Prescott’s and Hamilton–Adams’ interpolations (Figs. 7–16).

From Tables 1 and 2, on the basis of simple majority, Freeman’s algorithm outperforms the other algorithms. On the other hand, in two cases, it performs poorly.

For test image₁, as can be seen from Fig. 7, Linear interpolation produces the zipper effect that had been mentioned earlier. This is because linear interpolation is a low pass filter process and hence incorrectly locates the edges in each color plane, introducing zipper.¹² Cok’s interpolation reduces hue transitions over the edges since it interpolates the hue of the colors and not the colors themselves which reduces abrupt hue jumps producing fewer perceptual artifacts. Freeman’s algorithm, using the median as an estimator, performs poorly because it first performs a linear

Table 1 ΔE_{ab}^* errors for different interpolation algorithms after demosaicking.

| Algorithm used | Test image ₁ | Test image ₂ | Test image ₃ | Test image ₄ | Macaw ROI ₁ | Macaw ROI ₂ | Crayon ROI ₁ | Crayon ROI ₂ |
|------------------|-------------------------|-------------------------|-------------------------|-------------------------|------------------------|------------------------|-------------------------|-------------------------|
| Linear | 34.731 | 65.487 | 57.553 | 9.711 | 15.457 | 23.299 | 7.293 | 3.645 |
| Cok | 16.352 | 27.122 | 30.828 | 11.437 | 11.017 | 14.924 | 6.003 | 4.131 |
| Freeman | 15.179 | 55.301 | 19.513 | 9.599 | 5.404 | 7.421 | 4.649 | 3.645 |
| Laroche–Prescott | 7.321 | 0 | 24.592 | 10.944 | 11.028 | 14.198 | 5.507 | 4.234 |
| Hamilton–Adams | 3.052 | 0 | 21.793 | 9.303 | 9.279 | 11.579 | 4.409 | 3.936 |

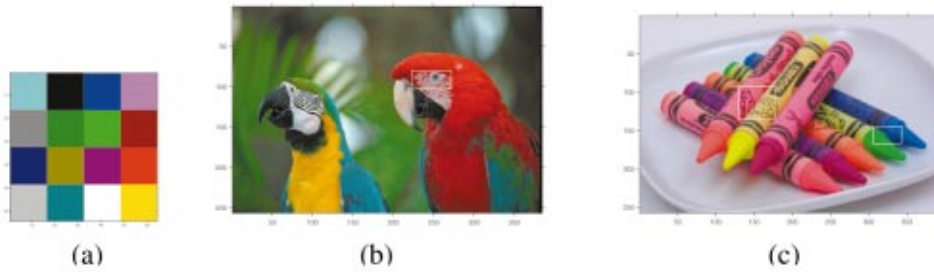


Fig. 6 Type II images: (a) test image₄, (b) original RGB Macaw image showing ROIs, and (c) original Crayon image showing ROIs.

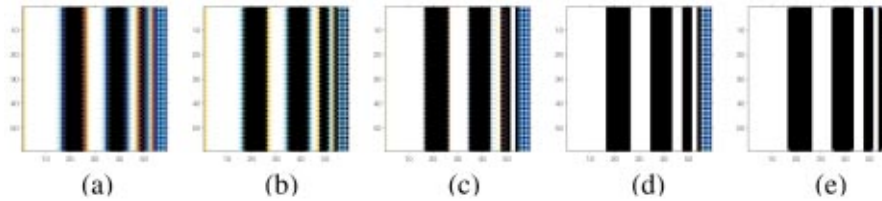


Fig. 7 (a) Linear, (b) Cok, (c) Freeman, (d) Laroche–Prescott, (e) Hamilton–Adams interpolations on test image₁. Note: Images are not the same size as original. Image has been cropped to hide edge effects.

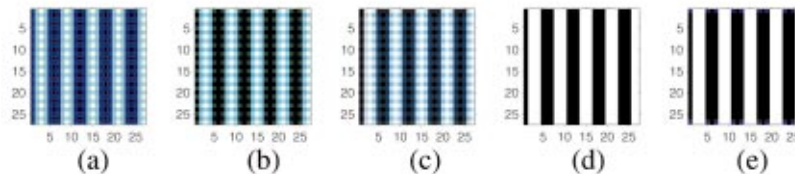


Fig. 8 (a) Linear, (b) Cok, (c) Freeman, (d) Laroche–Prescott, (e) Hamilton–Adams interpolations on test image₂. Note: Images are not the same size as original. Image has been cropped to hide edge effects.

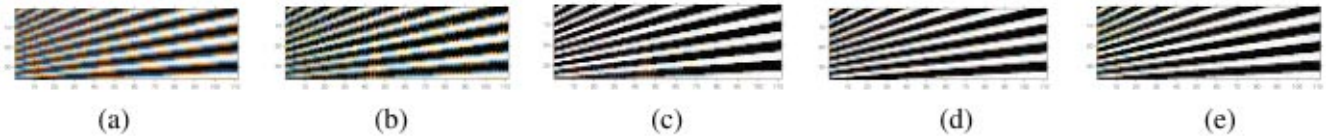


Fig. 9 (a) Linear, (b) Cok, (c) Freeman, (d) Laroche–Prescott, (e) Hamilton–Adams interpolations on test image₃. Note: Images are not the same size as original. Image has been cropped to hide edge effects.

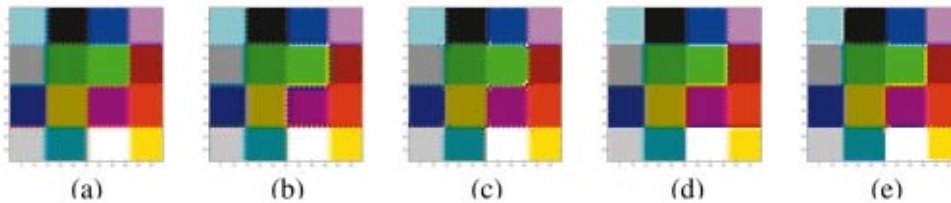


Fig. 10 (a) Linear, (b) Cok, (c) Freeman, (d) Laroche–Prescott, (e) Hamilton–Adams interpolations on test image₄. Note: Images are not the same size as original. Image has been cropped to hide edge effects.

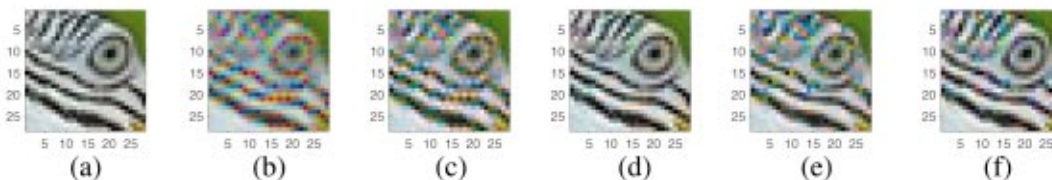


Fig. 11 (a) Original “truth” ROI₁ of Macaw image, (b) Linear, (c) Cok, (d) Freeman, (e) Laroche–Prescott, (f) Hamilton–Adams interpolations on Macaw image. Note: Images are displayed along with original image for comparison purposes.

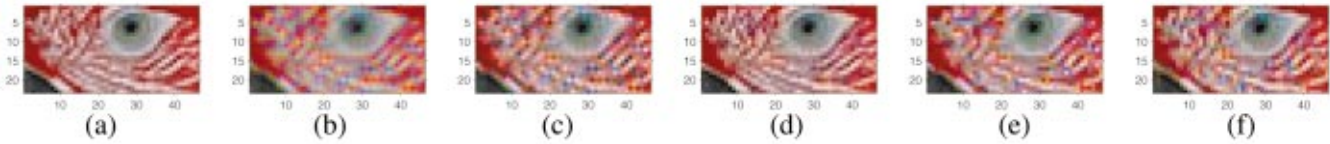


Fig. 12 (a) Original truth ROI_2 of Macaw image, (b) Linear, (c) Cok, (d) Freeman, (e) Laroche–Prescott, (f) Hamilton–Adams interpolations on Macaw image. Note: images are displayed along with original image for comparison purposes.

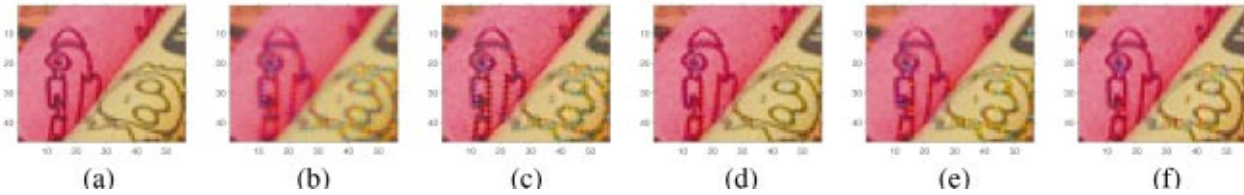


Fig. 13 (a) Original truth ROI_1 of Crayon image, (b) Linear, (c) Cok, (d) Freeman, (e) Laroche–Prescott, (f) Hamilton–Adams interpolations on Macaw image. Note: Images are displayed along with original image for comparison purposes.

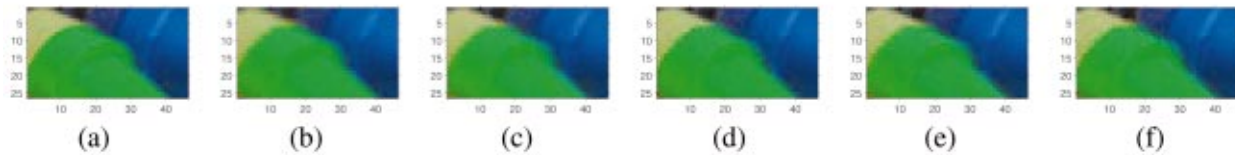


Fig. 14 (a) Original truth ROI_2 of Crayon image, (b) Linear, (c) Cok, (d) Freeman, (e) Laroche–Prescott, (f) Hamilton–Adams interpolations on Macaw image. Note: Images are displayed along with original image for comparison purposes.

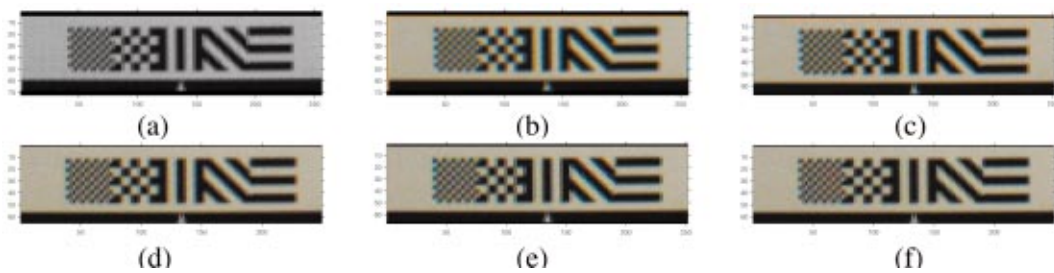


Fig. 15 (a) Original image CFA_1 , (b) Linear, (c) Cok, (d) Freeman, (e) Laroche–Prescott, (f) Hamilton–Adams interpolations.

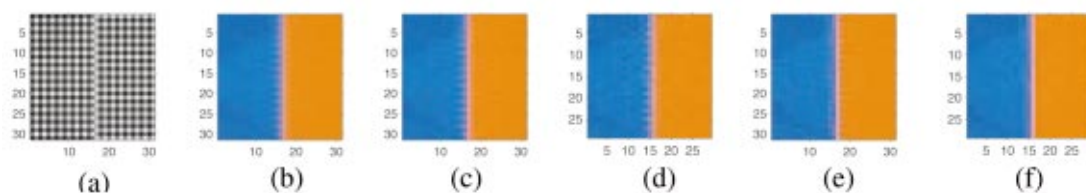


Fig. 16 (a) Original image CFA_2 , (b) Linear, (c) Cok, (d) Freeman, (e) Laroche–Prescott, (f) Hamilton–Adams interpolations.

Table 2 MSE ($\times 10^{-3}$) for different interpolation algorithms after demosaicking.

| Algorithm used | Test image ₁ | Test image ₂ | Test image ₃ | Test image ₄ | Macaw ROI ₁ | Macaw ROI ₂ | Crayon ROI ₁ | Crayon ROI ₂ |
|------------------|-------------------------|-------------------------|-------------------------|-------------------------|------------------------|------------------------|-------------------------|-------------------------|
| Linear | 154 | 253 | 101.6 | 18.1 | 33.0 | 68.6 | 10.4 | 1.7 |
| Cok | 100 | 163 | 67.3 | 31.0 | 20.5 | 37.5 | 6.7 | 2.1 |
| Freeman | 52.2 | 134 | 5.7 | 19.9 | 3.9 | 3.4 | 2.8 | 1.6 |
| Laroche–Prescott | 35.3 | 0 | 8.8 | 26.2 | 20.1 | 31.5 | 5.8 | 1.9 |
| Hamilton–Adams | 21.4 | 0 | 8.3 | 26.6 | 11.7 | 10.5 | 3.3 | 1.9 |

interpolation for the green channel (a blur process), also introducing ripples. Laroche–Prescott’s algorithm, using classifiers to interpolate in the preferred orientation reduces errors. Also, interpolating color differences (chrominance minus luminance), it utilizes information from two channels to precisely locate the edge. Hamilton–Adams’ algorithm interpolates the luminance channel with a bias to the second derivative of the chrominance channel, locating the edge in the three color planes with better accuracy.

In test image₂, although we find the same trend in Linear and Cok interpolations as we did in test image₁, we find that Laroche–Prescott’s and Hamilton–Adams’ algorithms are able to reproduce the image exactly. This is attributed to the structure (and size) of their estimators and the width of the bars themselves (three pixels).

In test image₃, there are two factors that the algorithms are tested against, one is varying spatial frequencies and the other being noncardinal edge orientations. Comparing Figs. 7 and 8 with Fig. 9, we observe that vertical and horizontal directions are reproduced with good clarity while edges along other orientations are not, alluding to the fact that almost all these algorithms (with the exception of Hamilton–Adams’, which incorporates some diagonal edge information) are optimized for horizontal and vertical edge orientations. A similar observation is made for the CFA images.

Note that in test image₄, the edge between the two green patches has been estimated with good accuracy by Laroche–Prescott’s and Hamilton–Adams’ algorithms. This is attributed to the fact that these two algorithms, unlike the others, use data from all the color planes for estimation. In this case, the data on either side of the edge being “similar,” the estimate was correct.

Another trend observed is that Hamilton–Adams’ algorithm performs better than Laroche–Prescott’s algorithm. This is attributed to two reasons; one that the process of estimating the green channels in Hamilton–Adams’ algorithm incorporates the second order gradient in the chrominance channels also, providing a better estimate while Laroche–Prescott’s algorithm simply performs a preferential averaging. The second reason is that Hamilton–Adams’ algorithm estimates diagonal edges while estimating the chrominance channels, giving it more sensitivity to noncardinal chrominance gradients (which partially explains the slightly smaller error metrics for test image₃).

6 Conclusion

It has been demonstrated that although the CFA pattern is useful to capture multispectral data on a monolithic array, this system comes with associated problems of “missing samples.” The estimation of these missing samples needs to be done in an efficient manner, at the same time, reproducing the original images with high fidelity.

In general, we observe two types of error zipper effect errors (occur at intensity edges see Fig. 7 for this behavior) confetti errors (occur at bright pixels surrounded by a darker neighborhood see Figs. 12 and 11 for this behavior). Experimentally, it has been found that Freeman’s algorithm is best suited for cases in which there is speckle behavior in the image, while Laroche–Prescott’s and Hamilton–Adams’ algorithms are best suited for images with sharp edges.

It is to be noted that demosaicking is not shift invariant. Different results are observed if the location of the edges is phase shifted (the zipper effect errors show up either as blue-cyan errors or as orange-yellow errors depending upon edge-location, see Fig. 7). The result of demosaicking is, hence, a function of the edge location.

Acknowledgments

The authors would like to thank the Army Research Office for its support in this work. This work is the first step in the development of a set of rugged, robust multispectral sensors for Army applications. We are also grateful to Pulnix America Inc. for providing us with a camera for this project.

Appendix: XYZ to CIELAB Conversion

Two of the color models suggested by the CIE which are perceptually balanced and uniform are the CIELUV and the CIELAB color models. The CIELUV model is based on the work by MacAdams on the just noticeable differences in color.¹⁶ These color models are nonlinear transformations of the XYZ color model. The transformation from the XYZ space to the CIELAB space is given by

$$L^* = \begin{cases} 116 \left(\frac{Y}{Y_n} \right)^{1/3} - 16 & \text{for } \frac{Y}{Y_n} > 0.008856 \\ 903.3 \left(\frac{Y}{Y_n} \right) & \text{otherwise} \end{cases},$$

$$a^* = 500 \left[\left(\frac{X}{X_n} \right)^{1/3} - \left(\frac{Y}{Y_n} \right)^{1/3} \right],$$

$$b^* = 200 \left[\left(\frac{Y}{Y_n} \right)^{1/3} - \left(\frac{Z}{Z_n} \right)^{1/3} \right],$$

where X_n , Y_n , Z_n are the values of X , Y , Z , for the appropriately chosen reference white; and where, if any of the ratios (X/X_n) , (Y/Y_n) , or (Z/Z_n) is less than or equal to 0.008 856, it is replaced in the above formula by $7.787F + 16/116$ where F is (X/X_n) , (Y/Y_n) , or (Z/Z_n) as the case may be. The color differences in the CIELAB color space are given by $\Delta E_{ab}^* = \sqrt{(\Delta L^*)^2 + (\Delta a^*)^2 + (\Delta b^*)^2}$.

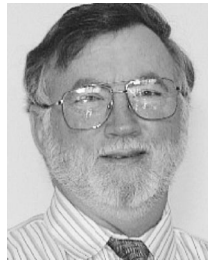
References

1. B. E. Bayer, "Color imaging array," U.S. Patent No. 3,971,065 (1976).
2. D. R. Cok, "Signal processing method and apparatus for producing interpolated chrominance values in a sampled color image signal," U.S. Patent No. 4,642,678 (1987).
3. W. T. Freeman, "Median filter for reconstructing missing color samples," U.S. Patent No. 4,724,395 (1988).
4. C. A. Laroche and M. A. Prescott, "Apparatus and method for adaptively interpolating a full color image utilizing chrominance gradients," U.S. Patent No. 5,373,322 (1994).
5. J. F. Hamilton and J. E. Adams, "Adaptive color plane interpolation in single sensor color electronic camera," U.S. Patent No. 5,629,734 (1997).
6. R. Kimmel, "Demosaicking: Image reconstruction from color ccd samples," *IEEE Trans. Image Process.* **7**(3), 1221–1228 (1999).
7. H. J. Trussell, "Mathematics for demosaicking," *IEEE Trans. Image Process.* (to be published).
8. R. Ramanath, "Interpolation methods for the bayer color array," MS thesis, North Carolina State University, Raleigh, NC (2000).
9. P. L. Vora, J. E. Farrell, J. D. Teitz, and D. H. Brainard, "Digital color cameras-1-response models," Hewlett-Packard Laboratory Technical Report, No. HPL-97-53 (1997).
10. G. Bilbro and W. E. Snyder, "Optimization by mean field annealing," *Advances in Neural Information Processing Systems* **1**, 91–98 (1989).
11. J. G. Proakis and D. G. Manolakis, *Digital Signal Processing—Principles, Algorithms and Applications*, 3rd. ed., Prentice Hall, Englewood Cliffs, NJ (1998).
12. J. E. Adams, "Interactions between color plane interpolation and other image processing functions in electronic photography," *Proc. SPIE* **2416**, 144–151 (1995).
13. K. Topfer, J. E. Adams, and B. W. Keelan, "Modulation transfer functions and aliasing patterns of cfa interpolation algorithms," *IS&T PICS Conference*, pp. 367–370 (1998).
14. J. E. Adams, "Design of practical color filter array interpolation algorithms for digital cameras," *Proc. SPIE* **3028**, 117–125 (1997).
15. WD4 of ISO 17321, "Graphic technology and photography—Color characterization of digital still cameras using color targets and spectral illumination" (1999).
16. G. Wyszecki and W. S. Stiles, *Color Science—Concepts and Methods, Quantitative Data and Formulae*, 2nd ed., Wiley, New York (1982).
17. M. L. Mahy, V. Eyckdenm, and A. Oosterlinck, "Evaluation of uniform color spaces developed after the adoption of cielab and cielv," *Color Res. Appl.* **19**(2), 105–121 (1994).
18. R. C. Gonzalez and R. E. Woods, *Digital Image Processing*, Addison Wesley, Reading, MA (1992).



search interests include restoration techniques in image processing, demosaicking in digital color cameras, color science, and automatic target recognition.

Rajeev Ramanath (student member '00) received his BE degree in electrical and electronics engineering from Birla Institute of Technology and Science, Pilani, India in 1998. He obtained his ME degree in electrical engineering from North Carolina State University in 2000. His Masters thesis was titled "Interpolation Methods for Bayer Color Arrays." Currently, he is in the doctoral program in electrical engineering at North Carolina State University. His research



Wesley E. Snyder received his BS in electrical engineering from North Carolina State University in 1968. He received his MS and PhD at the University of Illinois, also in electrical engineering. In 1976, Dr. Snyder returned to NCSU to accept a faculty position in electrical engineering, where he is currently a full professor. He served as a founder of the IEEE TAB Robotics Committee, which became the Robotics and Automation Society. He is sole

author of the first engineering textbook on robotics. Dr. Snyder then served as founder of the IEEE TAB Neural Networks Committee, which became the IEEE Neural Networks Council, and served in many administrative positions, including vice president. His research is in the general area of image processing and analysis. He has been sponsored by NASA for satellite-based pattern classification research, by NSF for robotic control, by the Department of Defense for automatic target recognition, by the West German Air and Space agency for spaceborne robot vision, and for a variety of industrial applications. He also has a strong interest in medical applications of this technology, and spent three years on the radiology faculty at the Bowman Gray School of Medicine. At NCSU, he is currently working on new techniques in mammography, inspection of integrated circuits, and automatic target recognition. He also has an appointment at the Army Research Office, in the areas of image and signal processing and information assurance. He is currently on the executive committee of the automatic target recognition working group. He has just completed a new textbook on machine vision.



computer engineering. He has published in image analysis, global optimization, neural networks, microwave circuits, and device physics. His current interests include analog integrated circuits and cathode physics.

Griff L. Bilbro received his BS degree in physics from Case Western Reserve University in Cleveland, Ohio, and his PhD degree in 1977 from the University of Illinois at Urbana-Champaign, where he was a National Science Foundation graduate fellow in physics. He designed computer models of complex systems in industry until 1984 when he accepted a research position at North Carolina State University. He is now a professor of electrical and computer engineering. He has published in image analysis, global optimization, neural networks, microwave circuits, and device physics. His current interests include analog integrated circuits and cathode physics.



William A. Sander III joined the U.S. Army research office (ARO) in 1975. The ARO is now part of the U.S. Army Research Laboratory. Currently he is the ARO associate director for computing and information science and directs an extramural research program including information processing, information fusion, and circuits. He has served as the Army representative on the joint services electronics program and as associate director of the electronics division. He has also served ARO as manager of command, control, and communications systems in the office of research and technology integration and as a program manager for signal processing, communications, circuits, and CAD of ICs in the electronics division. From 1970 until 1972, Dr. Sander was on active duty as a test project officer for the Mohawk OV1-D surveillance systems with the U.S. Army airborne, communications-electronics test board and served as a civilian in the position of test methodology engineer with the same organization until joining the Army research office in 1975. During the period 1989–1992, he served several extended detail assignments with the office of assistant secretary of the Army (research, development, and acquisition), the Army science board, and the office of the DoD comptroller. Dr. Sander received his BS degree in electrical engineering from Clemson University, Clemson, SC in 1964 and his MS and PhD degrees in electrical engineering from Duke University, Durham, NC in 1967 and 1973, respectively.