Journal of Electronic Imaging 12(4), 633–642 (October 2003).

Adaptive demosaicking

Rajeev Ramanath
Wesley E. Snyder
North Carolina State University
Department of Electrical and Computer Engineering
Box 7914
Raleigh, North Carolina 27695-7914
E-mail: rajeev.ramanath@ieee.org

Abstract. *Digital still color cameras sample the visible spectrum using an array of color filters overlaid on a CCD, 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 a value of a color band not sampled at a certain location is estimated from its neighbors. This is often referred to as "demosaicking." The human retina has a similar structure, although the distribution of cones is not as regular. Motivated by the human visual system, we propose an adaptive demosaicking technique in the framework of bilateral filtering. This approach provides us with a means to denoise, sharpen, and demosaic the image simultaneously. The proposed method, along with a variety of existing demosaicking strategies, are run on synthetic images and real-world images for comparative purposes. A recently proposed image comparison measure geared specifically toward demosaicking has also been applied to these images to provide a performance measure.*

[DOI: 10.1117/1.1606459]

1 Introduction

Commercially available digital still color cameras (DSC) are based on a single CCD array and capture color information by using color filters, each pixel capturing only one sample of the color spectrum. It has been more than 25 years since the invention patented by Bayer on the color filter array (CFA)—by far the most popular color filter array tessellation.[1] As an illustration, we have shown in Fig. 1 (adapted from Ref. 2), the Bayer CFA and a rendering of the cones in the human retina. Unlike the human retina, the CFA has a regular tessellation of color filters. However, both systems need to perform similar tasks with regard to generating a full color composite of the scene. We need to bear in mind though that the density and arrangement of the cones in the retina is not the same for all individuals—the nervous system, however, does a remarkable job of learning how to generate a full color image from such a pattern.

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). In the Bayer array, at each pixel location only one spectral band is sampled. The mosaic of color samples thus generated needs to be populated with information from all the color planes to obtain a full-resolution color image. This process is often referred to as demosaicking. We have the following image formation model

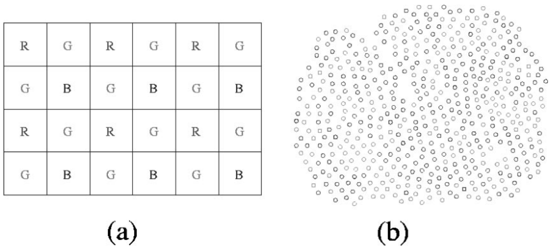
$$g(m,n)=\mathcal{B}\{\mathcal{H}\otimes f(m,n,o)+\mathcal{N}\},\quad o=1,2,3, \tag{1}$$

where $g(.)$ is the mosaicked image formed, $\mathcal{B}$ is a Bayer subsampling operator, $\mathcal{H}$ is a blur operator, $\otimes$ represents the spatial convolution operator, $f(.)$ is the original image (with multiple channels indexed by o), and $\mathcal{N}$ is representative of additive white noise. It is assumed that the noise is not channel dependent, and the blur is the same for all the channels under consideration. The demosaicking algorithms attempt at inverting $\mathcal{B}$. In essence, demosaicking is a type of interpolation, but attempts at minimizing the artifacts that may result from conventional interpolation techniques.

It is a similar situation in the human eye, where a host of complex neurons (including retinal processing in the ganglia, bipolar, and amacrine cells) within and outside the eye, help undo this mosaic to create a full color image. Interestingly though, it took scientists many years before theories that were relatively well established in the realm of color science and vision were applied to demosaicking images obtained from CFAs.

In this document, we pose the problem of demosaicking as a bilateral filtering process to highlight its striking similarity to the retinal and postretinal processing in the human visual system (HVS). Bilateral filtering smoothes images while preserving edges, by means of nonlinear combinations of neighboring image pixel values.[3] A bilateral filter can enforce similarity measures such as squared or CIELAB errors between neighbors while performing the standard filtering operation.[4] There are few demosaicking algorithms in the open literature, the popular ones being the one proposed by Kimmel[5] and another by Gunturk, Altunbasak, and Mersereau.[6] The algorithm presented here is adaptive (the demosaicking kernel varies based on the strength of the edge at a given location) in its interpolation mechanism. We use a recently proposed[7] image comparison measure as a basis for comparing the performance of the proposed algorithm with popularly available techniques.

Paper 02094 received Oct. 1, 2002; revised manuscript received Apr. 4, 2003; accepted for publication May 15, 2003.
1017-9909/2003/$15.00 © 2003 SPIE and IS&T.



R	G	R	G	R	G
G	B	G	B	G	B
R	G	R	G	R	G
G	B	G	B	G	B

Fig. 1 Comparison of (a) Bayer color filter array and (b) the distribution of cones in the human retina.

In Sec. 2, we highlight some of the important achievements in the field of color science and their impact on our understanding of the human visual system. Section 3 puts these discoveries in the perspective of demosaicking. In Sec. 4 we introduce bilateral filtering. There is a wide variety of kernels that may be used in such a filter, some of which are presented in Sec. 5. In Sec. 6 we consider two important properties of bilateral filters that are ideally suited for the purpose of demosaicking. Section 7 presents modifications to the bilateral filtering process for the purpose of demosaicking. Results are presented in Sec. 8, performance measures in Sec. 9, and conclusions in Sec. 10. Parts of the work described here have been presented at recent conferences.[7,8]

2 Human Visual System

Young (in 1802) and von Helmholtz (in 1852) proposed the trichromacy theories of color, stating that human color experiences can be explained with three stimuli. Empirically, this theory stated that all colors in the visible spectrum may be matched by appropriate mixtures of three primary colors. This theory, although still valid, could not explain, for example, after-images. This later led to Hering (in 1874) proposing the opponent color theory, which still has a strong following in the research community. Hering based his work on the fact that certain colors appear to be linked together, introducing another level of computation in the HVS, the bipolar and ganglion cells. Color is perceived due to the relative activity of three kinds of opponents, red-green, yellow-blue, and black-white. This theory took color processing to a higher level to subsequent neural computing of signals.

It was not until Land's simple yet potent demonstration in 1961 with Mondrian patches that the theory, which is now popularly known as the retinex theory, became mainstay in explaining how we perceive colors.[9] The idea being that although there may be no spatial relationship between colors in the real world, our perception of these colors is dependent on how they are processed in the visual cortex of the brain. There is a spatial relationship between signals and the corresponding perception. Demosaicking techniques take advantage of this idea.

Receptive fields map the many millions of sensors in the eye onto about one million optic nerve fibers. In other words, they perform some sort of spatial encoding of signals. The idea of the receptive field was first brought forth by Hartline using extraordinarily simple technology in the 1930s.[10] Granit performed the first microelectrode-based recording of signals from the ganglion cells in the retina.[11] Their work led them both to a Nobel prize in physiology and medicine in 1967. The ganglion cells of primate retina have center-surround receptive fields. In other words, the center of the receptive field is excitatory while the surround is inhibitory. This inherently has the concept of a spatial derivative built in. Directionally selective retinal ganglion cells respond to stimuli moving in a preferred direction and are inhibited by stimuli moving in the opposite or if the stimulus is kept stationary, clearly demonstrating that certain orientations may be preferred for ganglion cells that are oriented in a certain way.[12]

After low-level retinal processing, the signals are processed by an even more sophisticated system in the visual cortex of the brain that has both color opponency and directional sensitivity. Hubel and Weisel's seminal experimental work in the 1960s led to a conclusive demonstration that primates respond selectively to edges at certain orientations.[13] This idea brings about the spatially opponent property of the primate vision system, combining the concept of receptive fields and the spatial relationship of neighboring cells in the visual cortex.

In 1985, Mullen found conclusive evidence proving that the HVS is more sensitive to luminance changes than chrominance changes.[14] He observed that a chromatic spatial grating yields a low-pass transfer function, while a luminance spatial grating yields a bandpass transfer function for the human observer, implying that we see luminance changes better than chrominance changes. This idea is exploited in the Bayer CFA and many image compression techniques, where the luminance channel is sampled at a much higher rate than the chrominance channels.

3 Comparison with Demosaicking

There is a variety of methods available to perform demosaicking, the simplest being linear interpolation. Linear interpolation obtains a linear estimate of the missing signals from the pixel neighbors. This, however, does not maintain edge information well. More sophisticated methods perform this interpolation by attempting to maintain edge detail[5,15–18] or limit hue transitions.[19] A comparative study of these algorithms has been performed in Refs. 20 and 21.

The following are properties of the HVS that are often used in performing demosaicking:

- three (or more) types of sensors
- color opponency (red versus green and yellow versus blue)
- center surround receptive fields
- reduced sensitivity to chrominance edges when compared to luminance
- increased sensitivity to vertical and horizontal orientations.

The Bayer array may be formulated using a multiplicity of colors in different phase arrangements and color filters. Although it is easy to conceptualize of a typical RG-GB arrangement, the color filters could be designed to be in cyan, green, yellow, and white (CGYW) arrangement also. It is to be noted, though, that the colors span a large subspace of the space of visible colors. The idea of opponency is used in some of the popular methods[15,19] in a very rudi-

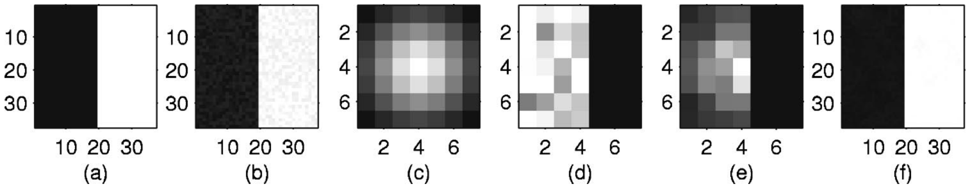


Fig. 2 Bilateral filtering: (a) original image, (b) image corrupted by Gaussian noise, (c) 7×7 blur kernel, (d) 7×7 similarity kernel at row=18, col=18, (e) 7×7 bilateral filter kernel and (f) resulting image (denoised and sharpened).

mentary form, where the green channel is considered to be representative of the luminance channel and the remaining sensor types define the other two axes of the 3-D color space (redness-greenness and blueness-yellowness).

Essentially, it is desired to interpolate along edges rather than across them as our visual system is highly edge sensitive. The interpolation performed during demosaicking is done by selective weighting of the pixels (as is done in a receptive field situation with spatial opponency). Bilateral filtering provides a framework in which such a computation may be performed by selectively weighting pixels based on their locations in an image.

4 Bilateral Filtering

Bilateral filtering smoothes images while preserving edges by means of nonlinear combinations of neighboring image pixel values. A bilateral filter can enforce similarity measures (such as squared error or error in the CIELAB space) between neighbors while performing typical filtering operations.

In a discrete representation, classical spatial domain filtering is represented as a convolution operation as

$$g(m,n)=\sum_{k=-L_1}^{L_1}\sum_{l=-L_2}^{L_2} g_o(m-k,n-l)h(k,l), \tag{2}$$

where $g_o(.)$ is the original image, $h(.)$ the convolution kernel, $g(.)$ the resulting image, and L_1 and L_2 are integers. For a lossless system, we have

$$\sum_{k=-L_1}^{L_1}\sum_{l=-L_2}^{L_2} h(k,l)=1. \tag{3}$$

The convolution kernel $h(k,l)$ can be a function of the geometric distance between the pixel at (m,n) and its neighbors. For example, a Gaussian blur function (low-pass filter) with zero mean and spread σ_h^2 written as

$$h(k,l)=h(r)=\frac{1}{\sqrt{2\pi}\sigma_h}\exp\left(-\frac{r^2}{2\sigma_h^2}\right), \tag{4}$$

where

$$r^2=k^2+l^2, \tag{5}$$

where r is a floating point number. Similarly, we may define a space-variant kernel, such that the value of the kernel elements is a function of the similarity between the various pixels in the neighborhood. We may construct a kernel, such that

$$s(m,n,k,l)=\Theta[g_o(m,n),g_o(k,l)], \quad k,l\in\aleph(m,n), \tag{6}$$

where $\aleph(m,n)$ represents the neighborhood of the image pixel location indexed by (m,n). The choice of the function Θ (similarity kernel) is application-dependent. One such choice for a similarity kernel is

$$\Theta[g_o(m,n),g_o(k,l)] \\ =\frac{1}{\sqrt{2\pi}\sigma_s}\exp\left\{-\frac{\Delta E^2[g_o(m,n),g_o(k,l)]}{2\sigma_s^2}\right\}, \tag{7}$$

where ΔE is a similarity measure described in the next section. Combining the two kernels $h(.)$ and $s(.)$, one may devise a kernel that is an element-by-element product of $h(.)$ and $s(.)$, i.e., we obtain the bilateral filter kernel given by

$$b(m,n,k,l)=h(k,l)s(m,n,k,l). \tag{8}$$

Using $b(.)$ from Eq. (8) in Eq. (2) we have

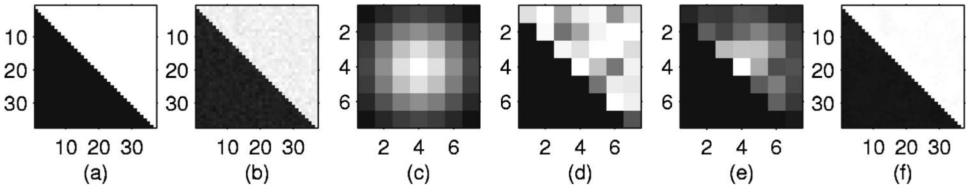


Fig. 3 Bilateral filtering: (a) original image, (b) image corrupted by Gaussian noise, (c) 7×7 blur kernel, (d) 7×7 similarity kernel at row=18, col=18, (e) 7×7 bilateral filter kernel, and (f) resulting image (denoised and sharpened).

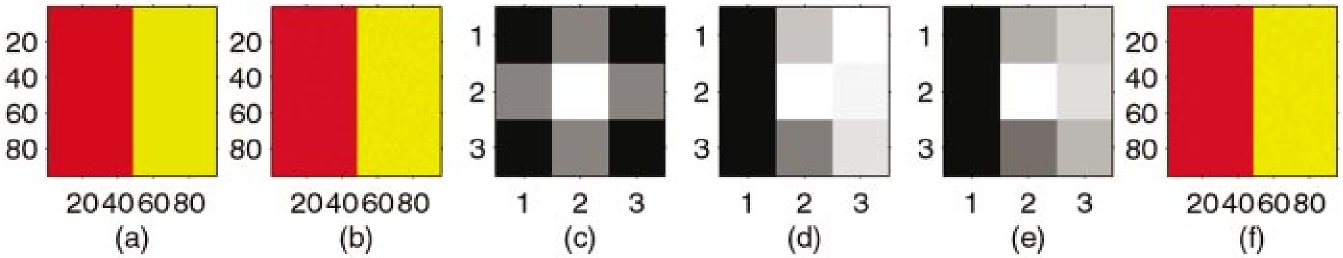


Fig. 4 Bilateral filtering using the ΔE^*_{ab} metric: (a) original image, (b) image corrupted by Gaussian noise, (c) blur kernel, (d) similarity kernel at row=50, col=51, (e) combined kernel, and (f) resulting image.

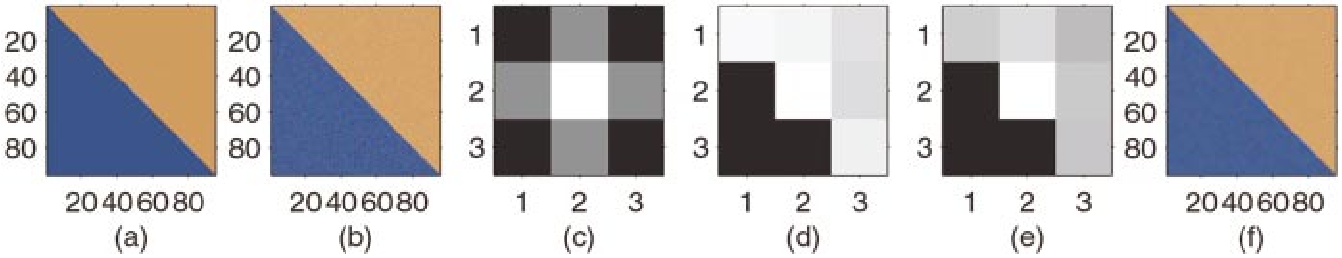


Fig. 5 Bilateral filtering using the ΔE^*_{ab} metric: (a) original image, (b) image corrupted by Gaussian noise, (c) blur kernel, (d) similarity kernel at row=50, col=51, (e) combined kernel, and (f) resulting image.

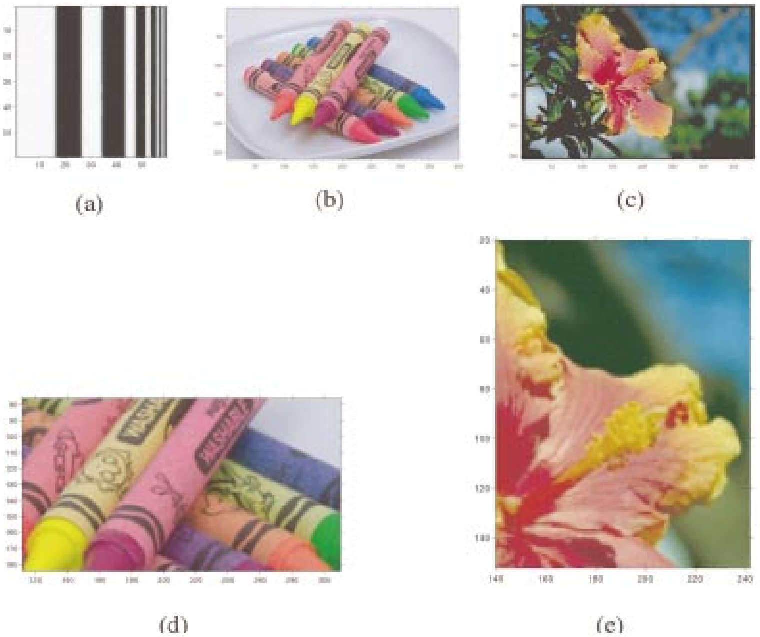


Fig. 11 Original images used in this experiment: (a) Test Image$_1$, (b) Crayon image, (c) Hibiscus image, (d) region of interest (ROI) in the Crayon image, and (e) ROI in the Hibiscus image.

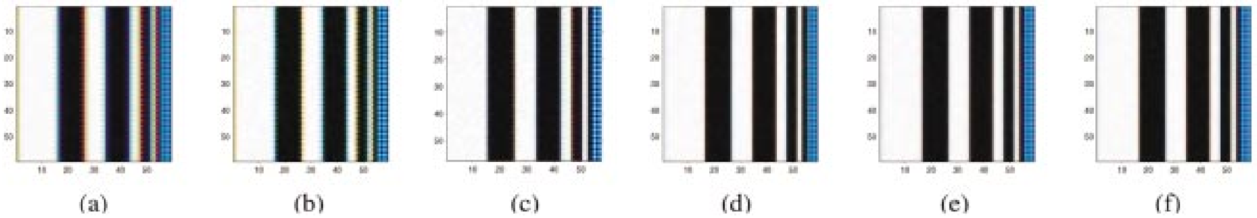
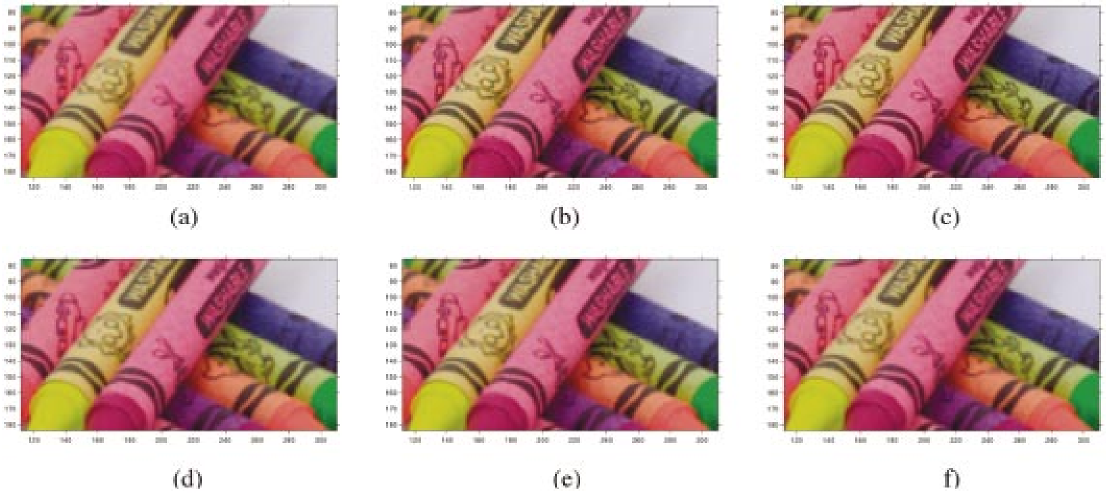
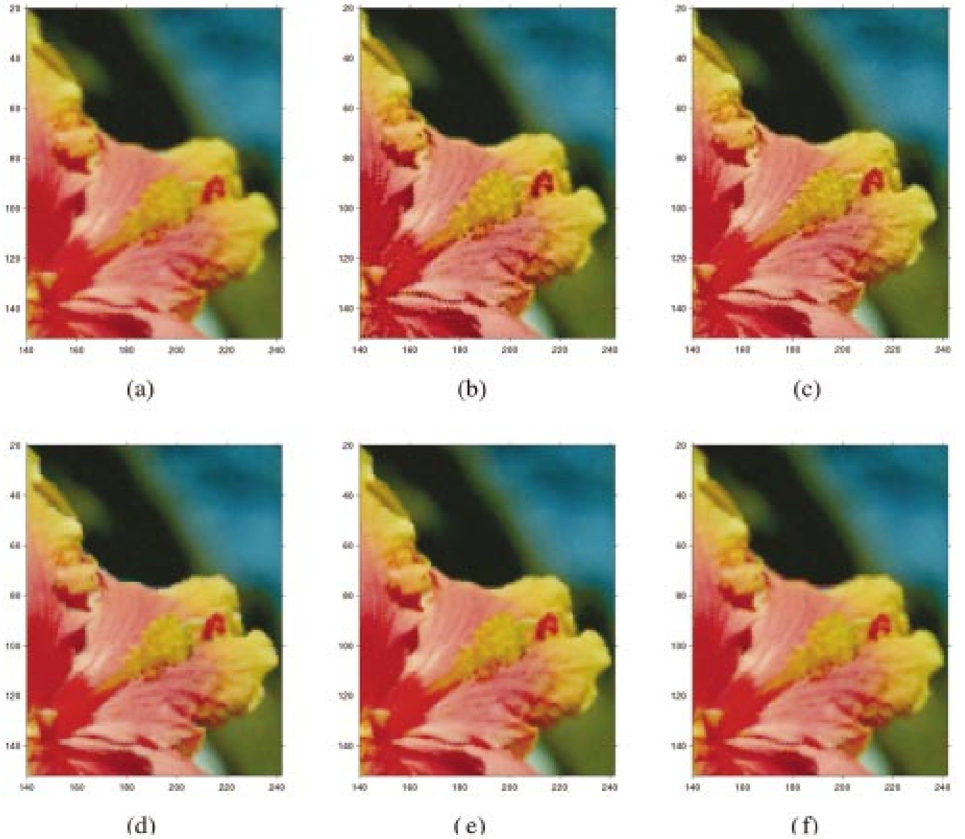


Fig. 12 Results using Test Image$_1$ with SNR of 30 dB: (a) linear, (b) Cok, (c) Freeman, (d) Bilateral$_1$ with σ_s=0.5, (e) Bilateral$_2$ with σ_s=0.5, and (f) Bilateral$_3$ with σ_s=20.


(a) (b) (c)

(d) (e) f)

Fig. 13 Results using Crayon image with SNR of 30 dB: (a) linear, (b) Cok, (c) Freeman, (d) Bilateral$_1$ with $\sigma_s = 0.5$, (e) Bilateral$_2$ with $\sigma_s = 0.5$, and (f) Bilateral$_3$ with $\sigma_s = 20$.


(a) (b) (c)

(d) (e) (f)

Fig. 14 Results using Hibiscus image with SNR of 30 dB: (a) linear, (b) Cok, (c) Freeman, (d) Bilateral$_1$ with $\sigma_s = 0.5$, (e) Bilateral$_2$ with $\sigma_s = 0.5$, and (f) Bilateral$_3$ with $\sigma_s = 20$.

$$g(m,n)=\sum_{k=-L_1}^{L_1}\sum_{l=-L_2}^{L_2} g_o(m-k,n-l)h(k,l)s(m,n,k,l), \tag{9}$$

where the product of $h(k,l)$ and $s(m,n,k,l)$ is normalized to satisfy

$$\sum_{k=-L_1}^{L_1}\sum_{l=-L_2}^{L_2} h(k,l)s(m,n,k,l)=1. \tag{10}$$

5 Various Kernel Combinations

We may choose any kernel for either of the two operations. Let us choose $h(k,l)$ as a Gaussian blur function with zero mean and spread σ_h^2, and $s(m,n,k,l)$ as a Gaussian penalty function (similarity kernel) with zero mean and spread σ_s^2. For grayscale images, ΔE may be chosen to be a function of the squared difference (SD) between pixel values.

$$\Delta E_1^2[g_o(m,n),g_o(k,l)]=[g_o(m,n)-g_o(k,l)]^2. \tag{11}$$

Figures 2 and 3 illustrate the performance of a 7×7 bilateral filter generated by using $h(.)$ as a Gaussian kernel with zero mean and $\sigma_h=2$, and $s(.)$ as a similarity kernel with $\sigma_s=1/20$. Figures 2(a) and 3(a) show the original images (of size 50×50), while Figs. 2(b) and 3(b) show the images corrupted by additive Gaussian noise. Figures 2(c) through 2(e) and 3(c) through 3(e) show the 7×7 blur kernel $h(.)$, the similarity kernel $s(.)$, and the resulting bilateral filter kernel $b(.)$ at pixel location (25,25), respectively. The image resulting from bilateral filtering is shown in Figs. 2(f) and 3(f) for each case.

For color images, we may use Euclidean distances in the RGB color space, which are known to be relatively poor estimates for color differences that are viewed by a human observer. The pixel values may also be transformed to the CIELAB color space and ΔE defined by

$$\begin{aligned}\Delta E_2^2[g_o(m,n),g_o(k,l)]=&[g_{o_L}(m,n)-g_{o_L}(k,l)]^2\\&+[g_{o_a}(m,n)-g_{o_a}(k,l)]^2\\&+[g_{o_b}(m,n)-g_{o_b}(k,l)]^2,\end{aligned} \tag{12}$$

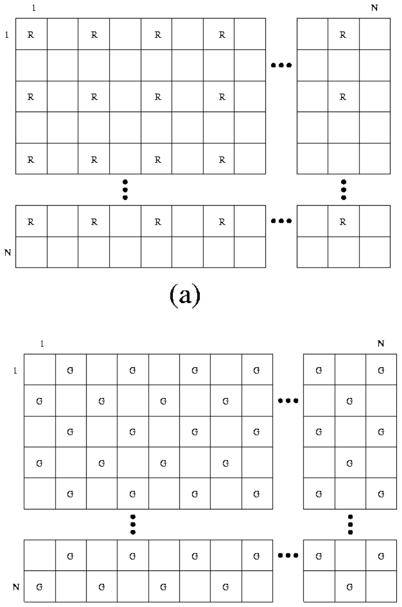
where subscripts L, a, and b correspond respectively to the CIELAB $L*$, $a*$, and $b*$ values of the two colors.

To use differences that correspond to perceived differences, we use ΔE to be the ΔE_{ab}^* error, the weighted Euclidean distance between colors in the CIELAB color space (relatively perceptually uniform). Figures 4 and 5 illustrate this for a bilateral filter of size 3×3.

6 Properties of the Bilateral Filter

Based on the experiments presented, let us look at the properties of the bilateral filter, denoted by $b(m,n,k,l)$.

- $b(m,n,k,l)$ may be defined for all edge orientations. This is because $h(k,l)$ has circular symmetry, while $s(m,n)$ adapts to edge orientation (as is shown in Figs. 2 and 3).



(a)

(b)

Fig. 6 Bayer array: (a) red channel and (b) green channel.

- If $h(k,l)$ is a blur function and $s(m,n,k,l)$ is as given in Eq. (7) we perform smoothing (denoising) as well as sharpening. $h(k,l)$ smoothes out noise, while $s(m,n,k,l)$ restricts smoothing across edges.

Both these properties are highly desirable for demosaicking. Current implementations of demosaicking methods used a fixed structure for interpolation, optimized for horizontal and vertical edge orientations only (clearly, this is driven by the compelling need for economical implementation in hardware). The suggested formulation circumvents that limitation. Unlike linear interpolation, such a technique would not perform low-pass operations across edges either. Moreover, the existing demosaicking techniques do not explicitly perform any denoising. Using the bilateral filtering scenario, denoising is performed implicitly.

7 Modifications for Demosaicking

The edge-sensitive filter kernels shown in Figs. 2(e) and 3(e) demonstrate a possibly useful extension to visualize the process of demosaicking. In a mosaicked image, the red channel (and similarly the blue) and the green channel may be visualized as shown in Figs. 6(a) and 6(b), respectively. The pixel locations with no entry may be regarded as zero valued. The algorithm presented here may be best explained with the help of an example. To estimate the green channel at the locations where green measurements were not made, one may adopt one of the following strategies: perform interpolation to estimate the missing samples; or perform edge-based interpolation to estimate the missing samples (most demosaicking algorithms use this strategy); or perform interpolation (linear, for simplicity) to estimate the bilateral filter kernel at each location and perform the proposed adaptive interpolation to estimate the missing samples.

Unlike the first strategy, the second and third make use of the edge information and may be expected to give better results. We use the third approach in this work. Let $\hat{f}(m,n,l)$ ($l=1$, 2, or 3, for the red, green, and blue channels—the resulting image is trivariate—RGB) be the desired estimated result, and $g(m,n)$ be the input image from the Bayer color array. Using the indexing shown in Fig. 6(b), for the green channel denoted by $\hat{f}(m,n,2)$, we have

$$\begin{aligned}\hat{f}(m,n,2)=&\hat{f}(m_{\text{odd}},n_{\text{even}},2)+\hat{f}(m_{\text{even}},n_{\text{odd}},2)\\&+\hat{f}(m_{\text{odd}},n_{\text{odd}},2)+\hat{f}(m_{\text{even}},n_{\text{even}},2),\\&\forall m,n\in[1...N],\end{aligned}\tag{13}$$

where the subscripts odd and even correspond to odd- and even-valued indices in the Bayer color array pixel locations for the rows and columns (m and n, respectively).

$$\begin{aligned}&\hat{f}(m_{\text{odd}},n_{\text{even}},2)=g(m_{\text{odd}},n_{\text{even}}),\\&\hat{f}(m_{\text{even}},n_{\text{odd}},2)=g(m_{\text{even}},n_{\text{odd}}),\\&\hat{f}(m_{\text{odd}},n_{\text{odd}},2)=\sum_{k=-L}^{L}\sum_{l=-L}^{L}g(m_{\text{odd}}-k,n_{\text{odd}}-l)\\&\qquad\times b'(m_{\text{odd}},n_{\text{odd}},k,l),\\&\hat{f}(m_{\text{even}},n_{\text{even}},2)=\sum_{k=-L}^{L}\sum_{l=-L}^{L}g(m_{\text{even}}-k,n_{\text{even}}-l)\\&\qquad\times b'(m_{\text{even}},n_{\text{even}},k,l),\end{aligned}\tag{14}$$

where $b'(.)$ is the normalized version of $b(.)$ and L is chosen to be a fixed integer value (note that the kernel is square). Equation (14) states that at the locations where we have green measurements, we do not perform any interpolation, and at the other locations, we perform bilateral filtering to estimate the samples. A similar set of equations can be written for the blue and red channels.

A well-known image formation model in demosaicking is to assume that the RGB color planes are perfectly correlated over the extent of the interpolation neighborhood.[22,23] In other words, the green pixel values in the demosaicked image is highly correlated to the red/blue pixel values in the mosaicked image. This is given by

$$\begin{aligned}&\hat{f}(m_{\text{odd}},n_{\text{odd}},2)=g(m_{\text{odd}},n_{\text{odd}})+k_{rg},\\&\hat{f}(m_{\text{even}},n_{\text{even}},2)=g(m_{\text{even}},n_{\text{even}})+k_{bg},\end{aligned}\tag{15}$$

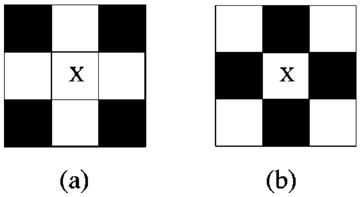


Fig. 7 Two types of neighbors (marked by a white pixel) for pixel marked with X: (a) four-neighbors and (b) corner neighbors.

where $\hat{f}(m_{\text{odd}},n_{\text{odd}},2)$ denotes a green sample at location $(m_{\text{odd}},n_{\text{odd}})$, where a red sample was measured; $\hat{f}(m_{\text{even}},n_{\text{even}},2)$ denotes a green sample at location $(m_{\text{even}},n_{\text{even}})$, where a blue sample was measured; and k_{rg} and k_{bg} are scalars. The estimate obtained from this assumption will clearly be inaccurate, but gives us a measure of the orientation of the edge (in other words, the variation of the pixel similarities over the neighborhood).

Let us define the four-neighbors and corner neighbors of a pixel. A pixel is a four-neighbor of a given pixel if they share a face, while a corner neighbor is one that shares only one vertex. This is shown in Fig. 7. Consider the case where we need to estimate the value of the green channel while at a red center. Referring to Fig. 1, the green samples are four-neighbors to the red sample. We use the mask in Fig. 7(a), along with the bilateral filter kernel, for this purpose. Similarly, to estimate the blue value at a red pixel, we use the mask in Fig. 7(b).

As an example, assume that we have a bilateral filter kernel [shown in Fig. 8(a)] at a red pixel. To obtain the demosaicking kernel for estimating the green channel, we use the bilateral filter kernel along with the mask in Fig. 8(b), to obtain the kernel shown in Fig. 8(c). This kernel is used in Eq. (14) to obtain the green estimate. Now, to estimate the blue sample at the same location, we use the mask as shown in Fig. 8(d), along with the bilateral filter kernel shown in Fig. 8(a) to obtain the kernel shown in Fig. 8(e). This kernel is now used to estimate the blue sample at a red location. The case for estimating the red and green pixels while at a blue pixel may be constructed in a similar manner.

At a green center, the case is much more simple, as it only requires two samples to be used in the mask. The two cases arising when the green pixel is on odd and even rows is illustrated in Figs. 9 and 10, respectively. The kernel to be used for estimating a red pixel when the measured sample is green (on an odd row index) is shown in Fig. 9(c), and that for a blue pixel is shown in Fig. 9(e).

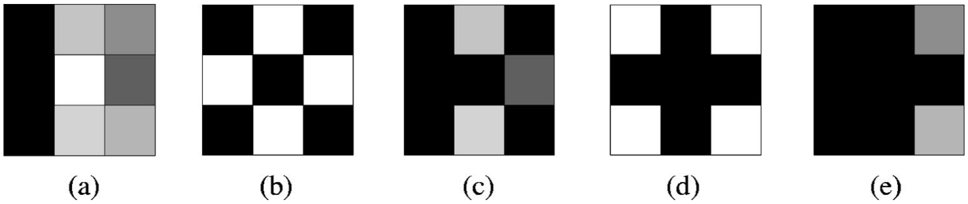


Fig. 8 Bilateral filtering kernels used for demosaicking at red center (similar for blue center): (a) kernel obtained using linearly interpolated green channel, (b) mask for green, (c) resulting kernel for green, (d) mask for blue, and (e) resulting kernel for blue.

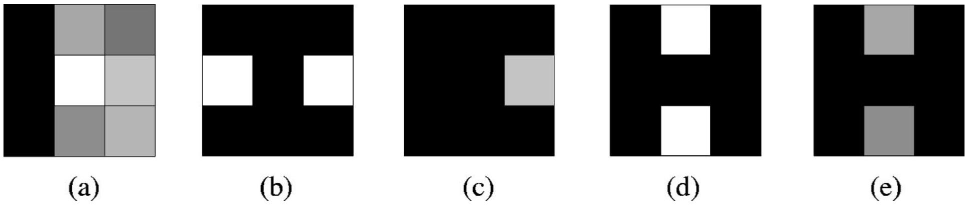


Fig. 9 Bilateral filtering kernels used for demosaicking at green center on odd rows: (a) kernel obtained using linearly interpolated green channel, (b) mask for red, (c) resulting kernel for red, (d) mask for blue, and (e) resulting kernel for blue.

Similarly, the kernel to be used to estimating a red pixel when the measured sample is green (on an even row index) is shown in Fig. 10(c), and that for a blue pixel is shown in Fig. 10(e). Using the resulting kernels, we may now perform bilateral filtering to estimate the missing values, providing a means to perform demosaicking that is sensitive to edges.

7.1 *Generating Bilateral Filter Kernels*

Earlier, we described how to modify the bilateral filter kernel for demosaicking. We have used three means to generate the bilateral filter kernel.

- Bilateral$_1$: Linear interpolate the green channel and use the squared difference between the linearly interpolated and the measured values [essentially providing us a measure of the deviation from the model in Eq. (15)] to generate the bilateral filter kernel.
- Bilateral$_2$: Linearly interpolate the green channel to arrive at an estimate and use the squared differences between neighbors using Eq. (11) to generate the bilateral filter kernel.
- Bilateral$_3$: Linearly interpolate all the channels and use the squared differences between the neighbors using Eq. (12) to generate the bilateral filter kernel.

8 Experimental Results

The previously mentioned techniques have been applied on one synthetic and two real-world images, shown in Fig. 11. The images are blurred with a low-pass Gaussian filter with spread (variance) of 0.2 and a kernel size of 3×3. Additive Gaussian noise with an SNR of 30 dB is added in an attempt to simulate the model described in Eq. (1). The result of various demosaicking algorithms on the synthetic and test images is shown in Figs. 12 and 13.

The results of the synthetic image (see Fig. 12) clearly demonstrate an improvement in the performance of demosaicking when compared to bilinear, Cok's, and Freeman's algorithms. The choice of σ_s will change the performance of the algorithm, however, its value does not need to be changed for different images. To address this, σ_s is kept constant for the three images. This gives us the ability to perform an unbiased comparison. Notice that performance of all algorithms is poor at high spatial frequencies, especially when the frequency is higher than the size of the demosaicking kernel. Also observe that Cok's and Freeman's algorithms do not perform any kind of noise removal.

In the Crayon image (see Fig. 13), Cok's algorithm severely enhances the noise in the image, while Freeman's algorithm introduces strong zipper artifacts (although it does produce sharp edges). The proposed method (and its variants) reconstructs the pink crayon reasonably well. The yellow crayon, however, shows zipper artifacts for all the algorithms (except Freeman's algorithm), especially in the region around the line art of a face on the crayon. This is attributed to a combination of the artifacts introduced by sampling using a Bayer CFA (edges occurring only over one pixel, sampling below the Nyquist rate) and the degree of blur (or antialiasing) in the original image.

In the Hibiscus image (see Fig. 14), the edges show considerable zipper artifacts in the bilinear interpolation. Cok's algorithm fails in regions that have low green intensities (regions that are saturated with red) because of the color ratios required by the algorithm that become large when green intensities are low. Freeman's algorithm also produces a few zipper artifacts. Bilateral$_1$ has reduced noise but the edges have considerable amount of zipper artifacts. Bilateral$_2$ has the best performance when compared with the other algorithms.

9 Performance Measures

A frequently used measure of performance is the squared error measure. Although this does not provide us much in-

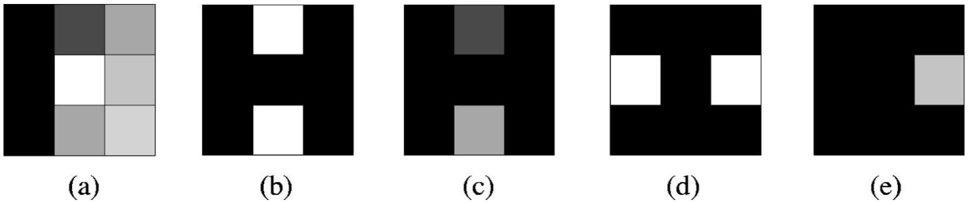


Fig. 10 Bilateral filtering kernels used for demosaicking at green center on even rows: (a) kernel obtained using linearly interpolated green channel, (b) mask for red, (c) resulting kernel for red, (d) mask for blue, and (e) resulting kernel for blue.

Table 1 Error measures for different demosaicking algorithms.

	Test image$_1$			Crayon image			Hibiscus image		
Algorithm used	MSE ($\times 10^{-3}$)	ΔE^*_{ab}	ΔE_1	MSE	ΔE^*_{ab}	ΔE_1	MSE	ΔE^*_{ab}	ΔE_1
Bilinear	137.8163	45.3278	28.7323	166.5634	27.6584	4.3878	262.2415	73.5687	4.5753
Cok	79.0262	33.7602	24.9654	110.7913	26.0156	3.9430	209.0626	66.3748	4.0611
Freeman	55.7876	30.9281	21.1265	117.2613	28.1720	3.4726	201.0862	64.6743	3.9670
Bilateral$_1$, $\sigma_s = 0.5$	39.0835	21.4198	14.6642	113.7406	26.5352	3.8060	196.2201	60.2472	3.8019
Bilateral$_2$, $\sigma_s = 0.5$	42.5876	21.0038	14.7559	106.2465	23.0984	3.5745	141.8772	58.0797	3.4921
Bilateral$_3$, $\sigma_s = 20$	41.9561	20.7784	15.9596	116.9842	29.1154	3.6201	200.1129	62.1183	3.8712

formation about the observed difference between images, it has a strong mathematical basis. Another measure is the error in the CIELAB space averaged over all the pixels in the image. Recently, the authors proposed an image comparison measure geared specifically toward images obtained from demosaicking algorithms.[7] The error measure is sensitive to spatial changes in hue, saturation, and chroma. From the results presented in Ref. 7, ΔE_1 using the ΔE_{00}[24] measure provides the best correlations with human observer results. It is given as:

$$\Delta E_1^2 = (1+\alpha)\Delta L^2 + (1+\beta)\Delta C^2 + (1+\gamma)\Delta H^2, \tag{16}$$

where α, β, and γ are weights based on the presence of an edge (an edge indicator of 0 or 1) in the luminance (L), chroma (C), and hue (H) channels of the image, respectively. In other words, the pixel values are transformed into the LCH domain and edge indicators calculated (may be performed earlier while determining the bilateral filter kernel). This measure should give us an idea of how well a given algorithm performs. These measures on the images resulting from each of the earlier mentioned algorithms are given in Table 1. Notice that three error measures are not always consistent in their grading from best to worst algorithms. However, given that ΔE^*_{ab} and ΔE_1 are both tested by observers, the latter being relatively more accurate than the former, this should provide a basis for judging the performance of the algorithms. These are tabulated in Table 1. Clearly, we see an improvement over popular techniques, supporting the claim that an adaptive demosaicking kernel does provide better estimates.

10 Conclusions

Depending on the quality of the reproductions of Figs. 12–14, one may be able to differentiate improvement in performance due to the proposed adaptive demosaicking. However, the measures in Table 1 should give us a better idea of their performance. It is, however, debatable if an observer would find consistent failure of an algorithm (as is seen with the bilinear algorithm) less objectionable than intermittent errors (as is seen with the proposed technique and some popular techniques).

An adaptive demosaicking technique is proposed and implemented. The algorithm presented here provides us with a means of choosing the interpolation kernel used for demosaicking. There are a lot of similarities between the human visual system and framework of bilateral filtering. Bilateral filtering provides us with a means to adaptively weight a pixel based on its location in an image. Although it is still unclear as to how the human visual system performs demosaicking, this work attempts at mimicking such processes. The extent to which we succeed is seen in the smaller error measures.

We need to bear in mind that the computational cost of this algorithm is high because of the need to generate the similarity kernel for every pixel at run-time. To this end, we suggest the use of look-up tables (LUTs) to speed-up the construction of the similarity kernel. This should bring in considerable increase in the computations and is proposed as further research.

Acknowledgments

The authors would like to thank the United States Army Space and Missile Defense Command for their support through grant number DASG60-03-1-0002. The authors also thank Phil Askey for the images used in this work.

References

1. B. E. Bayer, ‘‘Color imaging array,’’ U.S. Patent No. 3,971,065 (1976).
2. A. Roorda, A. Metha, P. Lennia, and D. R. Williams, ‘‘Packing arrangement of the three cone classes in the primate retina,’’ *Vision Res.* **41**(12), 1291–1306 (2001).
3. C. Tomasi and R. Manduchi, ‘‘Bilateral filtering for gray and color images,’’ *Sixth Int. Conf. Computer Vis.*, pp. 839–846 (1998).
4. G. Wyszecki and W. S. Stiles, *Color Science Concepts and Methods, Quantative Data and Formulae,* 2nd ed., John Wiley and Sons, Inc., New York (1982).
5. R. Kimmel, ‘‘Demosaicing: Image reconstruction from color ccd samples,’’ *IEEE Trans. Image Process.* **8**(9), 1221–1228 (1999).
6. B. K. Gunturk, Y. Altunbasak, and R. M. Mersereau, ‘‘Color plane interpolation using alternating projections,’’ *IEEE Trans. Image Process.* **11**(9), 997–1003 (2002).
7. R. Ramanath, W. E. Snyder, and D. Hinks, ‘‘Image comparison measure for digital stil color cameras,’’ *Proc. Int. Conf. Image Process.* **1**, 629–632 (2002).
8. R. Ramanath and W. E. Snyder, ‘‘Demosaicking as a bilateral filtering process,’’ *Proc. SPIE* **4667**, 236–244 (2002).
9. E. H. Land, ‘‘The retinex theory of color vision,’’ *Proc. Royal Inst. Great Britian* **47**, 23–58 (1974).
10. H. K. Hartline, ‘‘The receptive fields of optic nerve fibers,’’ *Am. J. Physiol.* **130**, 690–699 (1940).
11. R. Granit, *Sensory Mechanisms of the Retina*, Oxford University Press, London (1947).
12. H. B. Barlow and R. M. Hill, ‘‘Selective sensitivity to direction of movement in ganglion cells of the rabbit retina,’’ *Science* **139**, 412–414 (1963).
13. D. H. Hubel and T. N. Wiesel, ‘‘Receptive fields and functional architecture of monkey striate cortex,’’ *J. Physiol. (London)* **195**, 215–243 (1968).
14. K. T. Mullen, ‘‘The contrast sensitivity of human colour vision to red-green and blue-yellow chromatic gratings,’’ *J. Physiol. (London)* **359**, 381–400 (1985).

15. W. T. Freeman, ''Median filter for reconstructing missing color samples,'' U.S. Patent No. 4,724,395 (1988).
16. 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).
17. R. H. Hibbard, ''Apparatus and method for adaptively interpolating a full color image utilizing luminance gradients,'' U.S. Patent No. 5,382,976 (1995).
18. 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).
19. 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).
20. R. Ramanath, ''Interpolation methods for the bayer color array,'' MS Thesis, North Carolina State University, Raleigh, NC (2000).
21. R. Ramanath, W. E. Snyder, G. L. Bilbro, and W. A. Sander, ''Demosaicking methods for the bayer color array,'' *J. Electron. Imaging* **11**(3), 306–315 (2002).
22. J. E. Adams, ''Design of practical color filter array interpolation algorithms for digital cameras,'' *Proc. SPIE* **3028**, 117–125 (1997).
23. K. Topfer, J. E. Adams, and B. W. Keelan, ''Modulation transfer functions and aliasing patterns of cfa interpolation algorithms,'' *IS&T PICS Conf.*, pp. 367–370 (1998).
24. M. R. Luo, G. Cui, and B. Rigg, ''The development of the cie 2000 colour difference formula,'' *Color Res. Appl.* **26**(5), 340–350 (2001).

Rajeev Ramanath received his BE in electrical and electronic engineering from Birla Institute of Technology and Science, Pilani, India, in 1998. He obtained an MS in electrical engineering from North Carolina State University in 2000. Currently, he is in the doctoral program in electrical engineering at North Carolina State University and scheduled to graduate this year. His research interests include restoration techniques in image processing, demosaicking in digital color cameras, color science, and automatic target recognition.

Wesley E. Snyder received his BS in electrical engineering from North Carolina State University in 1968. He received the MS and PhD, also in electrical engineering. In 1976, he returned to NCSU to accept a faculty position in Electrical Engineering, where he is currently a full professor. His research is in the general area of image processing and analysis. 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, where he has supported the mission of that office for several years 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.