Adaptive Homogeneity-Directed Demosaicing Algorithm

Keigo Hirakawa, *Student Member, IEEE* and Thomas W. Parks, *Fellow, IEEE*

***Abstract*—A cost-effective digital camera uses a single-image sensor, applying alternating patterns of red, green, and blue color filters to each pixel location. A way to reconstruct a full three-color representation of color images by estimating the missing pixel components in each color plane is called a demosaicing algorithm. This paper presents three inherent problems often associated with demosaicing algorithms that incorporate two-dimensional (2-D) directional interpolation: misguidance color artifacts, interpolation color artifacts, and aliasing. The level of misguidance color artifacts present in two images can be compared using metric neighborhood modeling. The proposed demosaicing algorithm estimates missing pixels by interpolating in the direction with fewer color artifacts. The aliasing problem is addressed by applying filterbank techniques to 2-D directional interpolation. The interpolation artifacts are reduced using a nonlinear iterative procedure. Experimental results using digital images confirm the effectiveness of this approach.**

***Index Terms*—Color artifact, demosaicing algorithm, digital camera, filterbank, interpolation, metric neighborhood model.**

I. Introduction

In a typical digital camera, the optical image formed at the image plane is captured by a single CCD or CMOS sensor array [see Fig. 1(a)]. Each individual sensor is able to capture only a single color because of the arrangement of color films or dyes between the sensor and the lens. A demosaicing algorithm is a method for reconstructing a full three-color representation of color images by estimating the missing pixel components in each color plane. The arrangement of the color filter is called a color filter array or CFA. Fig. 1(b) shows the popular Bayer pattern [2]. The green pixels are sampled at a higher rate because the green color approximates the perceived brightness well, and the human eye is more sensitive to green compared to red and blue.

A demosaicing problem can be interpreted as a signal interpolation problem. Simple plane-wise interpolation, however, frequently results in color artifacts because the proportions of red, green, and blue are corrupted at object boundaries. Although it is a well-accepted fact that our visual systems are less sensitive to changes in color than changes in brightness, manifestation of colors utterly irrelevant to the scene stand out more than minor alterations in brightness. Furthermore, because it is often the case that like colors never appear adjacent to each other in a CFA, the output image often suffers from a pattern of alternating colors, commonly referred to as *zippering*. That is, the contribution of each color plane to the reconstruction of one pixel differs from that of its immediate neighboring pixel, consequently imposing an artificial repetitive pattern to the output. An example of this zippering atrifact appears in Fig. 10.

There are many algorithms proposed to overcome these difficulties by introducing color image models and exploiting structure between different color channels. For example, [4] and [11] use the property that the quotient of two color channels is slowly varying. This hypothesis follows from the fact that if two colors occupy the same coordinate in the chromaticity plane [16], then the ratios between the color components are equal. That is, let (r_1, g_1) and (r_2, g_2) be chromaticity coordinates for the tristimulus values (R_1, G_1, B_1) and (R_2, G_2, B_2), respectively. If $(r_1, g_1) = (r_2, g_2)$, then $R_1/G_1 = R_2/G_2$.

Alternatively, many assert that the differences between red, green, and blue images are slowly varying [1], [6], [7], [13]. This principle is motivated by the observation that the color channels are highly correlated. The difference image between green and red (blue) channels contains low-frequency components only, and, thus, the interpolation of the difference image becomes easier. A more sophisticated color channel correlation model is explored in [8].

Moreover, many demosaicing algorithms [1], [11], [12] incorporate edge directionality in interpolation. Interpolation along an object boundary is preferable to interpolation across this boundary for most image models.

Three inherent problems often associated with demosaicing algorithms that incorporate directional two-dimensional (2-D) interpolation are misguidance color artifacts, interpolation artifacts, and aliasing. The proposed demosaicing algorithm, which also adopts the directional interpolation approach, addresses these problems explicitly. In the process of estimating the missing pixel components, the aliasing problem is resolved by applying filterbank techniques to directional interpolation (Section II). This interpolation procedure produces two images: horizontally interpolated and vertically interpolated images. The level of misguidance color artifacts present in these images is compared by a color image homogeneity metric (Section III). The misguidance color artifacts are minimized by only keeping the pixels interpolated in the direction with fewer artifacts. The interpolation artifacts are reduced using a nonlinear iterative procedure (Section IV). By directing the algorithmic design to the artifact and alias problems explicitly, the proposed demosaicing algorithm achieves a significant improvement in the output image quality.

Manuscript received March April 8, 2003; revised January 17, 2004. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Bruno Carpentieri.

The authors are with Cornell University, Ithaca, NY 14850 USA (e-mail: kh237@cornell.edu).

Digital Object Identifier 10.1109/TIP.2004.838691

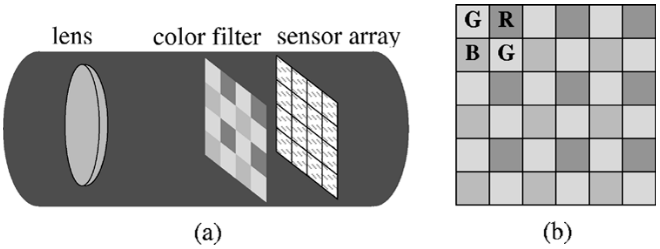


Fig. 1. (a) Digital optical system. (b) Bayer color filter array pattern.

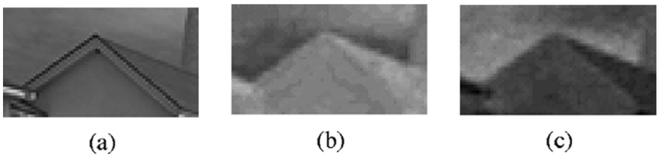


Fig. 2. Difference images. (a) Original image. (b) $R(x) - G(x)$. (c) $B(x) - G(x)$.

II. Alias and Bayer Pattern Interpolation

Let X be a set of 2-D pixel positions, and Y be a set of CIE RGB tri-stimulus values: $y \in Y$ then $y = [R, G, B]^T$, where R, G, and B represent red, green, and blue tri-stimulus values, respectively [14], [16]. Then a color image $f : X \rightarrow Y$ is a mapping between pixel locations and tri-stimulus values.

The goal in this section is to develop a technique to produce f_H and f_V, color images interpolated from CFA data using 2-D linear filters oriented in horizontal and vertical directions, respectively. Due to the rectangular sampling lattice, design of interpolation in horizontal and vertical directions is considerably simpler in the Bayer pattern than in other directions, such as diagonal [see Fig. 1(b)]. Only horizontal and vertical interpolations are explored in the proposed algorithm (see Section V-A for additional discussions).

The problem of estimating missing pixel components from CFA data can be interpreted as removing the aliasing terms from the sampled signals. In the proposed algorithm, filterbank theories are adapted in order to cancel the alias terms in the green channel by taking a linear combination of known pixels in the red and blue channels (Section II-A). The red and blue channels are reconstructed by removing the high-frequency components using the interpolated green channels (Section II-B).

A. Interpolation of Green Pixel Components

In this section, a method to reconstruct $G(\cdot)$ through horizontal interpolation is developed. Interpolation in the vertical direction is done in the same manner. Let $R(\cdot)$, $G(\cdot)$, and $B(\cdot)$ represent red, green, and blue color plane images, respectively, and $x \in X$. Assume $G(x) - R(x)$ is slowly varying [1], [6], [7], [13]. That is, the high-frequency components of the difference images decay more rapidly than those of $G(x)$: the power of $G(x) - R(x)$ at frequency $\xi = \pi/3$ is approximately one fifth of the power of $G(x)$ at the same frequency. Fig. 2 is an example of typical difference images, supporting this claim.

This subsection concentrates on estimating missing green pixels from known green and red pixel values using the green-red row of Bayer pattern (see Fig. 3). The same technique is used to estimating missing green pixels from known green and blue pixels. In the green-red row of Bayer pattern, the even samples of green image and the odd samples of red image are given. Given any signal $P(x)$, let $P_0(x)$ and $P_1(x)$ denote even and odd sampled signals of $P(x)$. That is

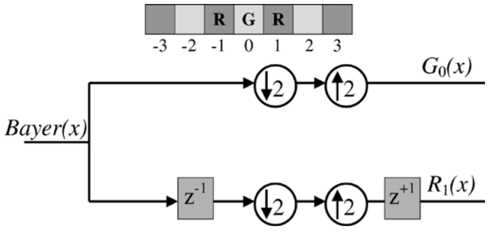


Fig. 3. Green-red row of Bayer array.

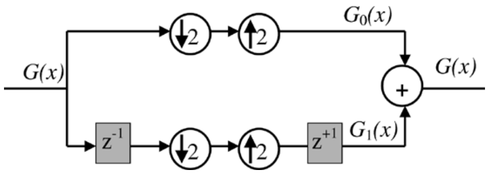


Fig. 4. Lazy wavelet.

$$P_0(x) = \begin{cases} P(x), & x \text{ even} \\ 0, & x \text{ odd} \end{cases} \qquad P_1(x) = \begin{cases} 0, & x \text{ even} \\ P(x), & x \text{ odd.} \end{cases}$$

For example, $G_0(x)$ and $G_1(x)$ are, respectively, even and odd-sampled signal of the image $G(x)$. As shown in Fig. 3, G_0 is available directly from the Bayer pattern, but G_1 is not. The relationship

$$G(x) = G_0(x) + G_1(x) \tag{1}$$

is known as a Lazy wavelet structure (see Fig. 4) [15]. To investigate how to obtain G_1, consider filtering G with a linear filter h. We would like h to have the property such that

$$G(x) = h(x) * G(x). \tag{2}$$

An example of $h(x)$ is an ideal low-pass filter, when G is band limited. Since the goal is to reconstruct the odd sampled-signal $G_1(x)$, substitute (1) into (2), and analyze it in terms of h_0 and h_1, the even and odd-sampled signals of h

$$\begin{aligned} G(x) = h_0(x) * G_0(x) + h_1(x) * G_0(x) \\ + h_0(x) * G_1(x) + h_1(x) * G_1(x). \end{aligned} \tag{3}$$

Due to the properties that

$$\begin{aligned} h_1(x) * G_0(x) &= 0, && \forall x \text{ even} \\ h_0(x) * G_1(x) &= 0, && \forall x \text{ even} \\ h_0(x) * G_0(x) &= 0, && \forall x \text{ odd} \\ h_1(x) * G_1(x) &= 0, && \forall x \text{ odd} \end{aligned}$$

the odd and even parts of (3) are

$$\begin{aligned} G(x) &= \begin{cases} h_0(x) * G_0(x) + h_1(x) * G_1(x), & x \text{ even} \\ h_1(x) * G_0(x) + h_0(x) * G_1(x), & x \text{ odd} \end{cases} \\ &= \begin{cases} G_0(x), & n \text{ even} \\ h_1(x) * G_0(x) + h_0(x) * G_1(x), & x \text{ odd.} \end{cases} \end{aligned} \tag{4}$$

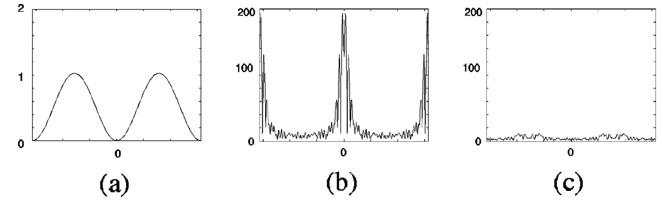


Fig. 5. Example of (5), absolute values are shown (a) $\hat{h}_0(\xi)$, (b) $\hat{G}_1(\xi) - \hat{R}_1(\xi)$, and (c) $\hat{h}_0(\xi) * \left(\hat{G}_1(\xi) - \hat{R}_1(\xi)\right)$.

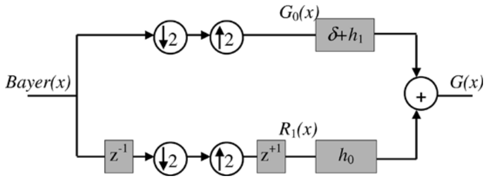


Fig. 6. Proposed method to estimating green image (6).

Note $h_0(x) * G_1(x)$ is still not available from the Bayer pattern. Therefore, $h_0(x)$ is chosen such that

$$h_0(x) * G_1(x) \approx h_0(x) * R_1(x). \tag{5}$$

Equivalently, $h_0(x)$ is designed to attenuate the sampled difference signal $G_1(x) - R_1(x)$. Fig. 5 shows an example using a horizontal slice from an image. The constraint (5) allows a substitution of (5) in (4)

$$G(x) = \begin{cases} G_0(x), & x \text{ even} \\ h_1(x) * G_0(x) + h_0(x) * R_1(x), & x \text{ odd} \end{cases}$$
$$= G_0(x) + h_1(x) * G_0(x) + h_0(x) * R_1(x). \tag{6}$$

Equation (6) is a method to estimate the green image from G_0 and R_1 only (see Fig. 6).

To design a zero-phase FIR filter $h(x)$ which meets our constraints (2) and (5), solve the following optimization problem:

$$\hat{h}_{\text{opt}}(\xi) = \arg\min_h \left\| \hat{w}(\xi)\left(1 - \hat{h}(\xi)\right)\right\|^2$$

where ˆ denotes the Fourier transform, $w(\cdot)$ is a weighting function, and h satisfies (5). For example, let $h(x)$ be length 5 and $w(\xi) = 2\pi - |\xi|$ (which weights the low-frequency regions more than in the high-frequency regions). To ensure that the constraint (5) is met, assume $\hat{h}_0(\xi = 0) = \hat{h}_0(\xi = \pi) = 0$. Using minimization tools in Matlab, we find

$$h_{\text{opt}}(x) = [-0.2569, 0.4339, 0.5138, 0.4339, -0.2569]. \tag{7}$$

Fig. 7(a) shows the frequency response of (7). From (7), the filters used in the proposed filterbank (Fig. 6) were derived: the frequency responses of $\delta(x) + h_1(x)$ and $h_0(x)$ are shown in Fig. 7(b) and (c), respectively. Fig. 7(b) illustrates that the low-frequency content of the output green image is taken from G_0. Fig. 7(c) illustrates that the condition in (5) is met.

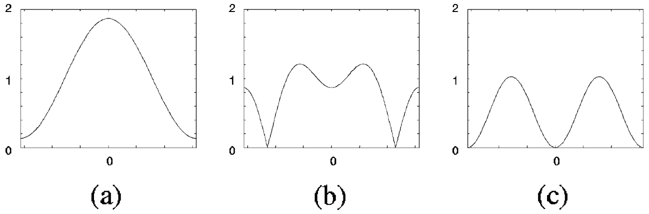


Fig. 7. Length 5 filter h_{opt}. (a) $\hat{h}(\xi)$. (b) $\hat{h}_1(\xi) + \hat{\delta}(\xi)$. (c) $\hat{h}_0(\xi)$

Let us summarize the green pixel image interpolation by examining the alias cancellation structure implicit in the filterbank (6). In the frequency domain, the filtered G_0 channel (the top channel in Fig. 6) takes the form

$$\begin{aligned} &\left(1 + \hat{h}_1(\xi)\right)\hat{G}_0(\xi) \\ &\quad = \left(1 + \frac{\hat{h}(\xi)}{2} - \frac{\hat{h}(\xi-\pi)}{2}\right)\left(\frac{\hat{G}(\xi)}{2} + \frac{\hat{G}(\xi-\pi)}{2}\right) \\ &\quad = \frac{1}{4\hat{G}(\xi)\left(3 - \hat{h}(\xi-\pi)\right)} + \frac{1}{4\hat{G}(\xi-\pi)\left(1 + \hat{h}(\xi)\right)}. \end{aligned} \tag{8}$$

Note that (2) is used to simplify some terms above. Using (5), the expansion of the filtered R_1 channel (the bottom channel) is

$$\begin{aligned} \hat{h}_0(\xi)\hat{R}_1(\xi) &= \hat{h}_0(\xi)\hat{G}_1(\xi) \\ &= \left(\frac{1}{2\hat{h}(\xi)} + \frac{1}{2\hat{h}(\xi-\pi)}\right)\left(\frac{1}{2\hat{G}(\xi)} - \frac{1}{2\hat{G}(\xi-\pi)}\right) \\ &= \frac{1}{4\hat{G}(\xi)(1 + \hat{h}(\xi-\pi))} - \frac{1}{4\hat{G}(\xi-\pi)(1 + \hat{h}(\xi))}. \end{aligned} \tag{9}$$

We may conclude that the alias terms in (8) are cancelled by the terms in (9).

B. Interpolation of Red/Blue Pixel Components

In this subsection (unlike in the previous subsection), we treat $G(\cdot)$, $R(\cdot)$, and $B(\cdot)$ as two dimensional signals.

The red pixel image R is reconstructed from the horizontally (vertically) interpolated green pixel image G [see (6)] and the known samples of red pixels (denoted R_S). Let G_S be the sampled signal of G at 2-D lattice locations corresponding to the known red pixels (i.e., rectangular lattice). Recall from the earlier hypothesis that the difference image $R - G$ is band limited at a rate well below the critical sampling rate. Therefore, the difference image $R - G$ is reconstructed from the sampled difference image $R_S - G_S$ using

$$R - G = L * (R_S - G_S) \tag{10}$$

where L is a 2-D low-pass filter. Using G from (6), solve (10) for R. The blue pixel image B is reconstructed using the same technique.

We remind the readers that red (blue) pixel image reconstructed using horizontally interpolated green pixel image is different from the red (blue) pixel image reconstructed using vertically interpolated green pixel image. Our final output color image $f_H(f_V)$ refers to the horizontally (vertically) interpolated green pixel image and red and blue pixel images reconstructed using this green image.

III. Misguidance Color Artifacts and Homogeneity

The goal in this section is to develop a method to compare the level of color artifacts present in f_H and f_V. The most major source of error in demosaicing occurs when the direction of interpolation is *erroneously* selected. In this paper, we call this phenomenon *misguidance* color artifact. We refer to color artifacts that surface for any other reasons as *interpolation* artifacts. A method to reduce interpolation artifacts is discussed in a later section.

Misguidance color artifacts are objectionable to the human eye, and we are interested in reducing them. Section III-A develops the concept of metric neighborhood models and homogeneity maps. Section III-B describes the relationship between homogeneity map and color artifacts. We show that a homogeneity map (13) can be used to analyze and minimize the level of misguidance color artifacts.

A. Metric Neighborhood Model Review

Metric neighborhood modeling offers a systematic method to identify a group of pixels that are similar [5], [10]. In this paper, similarity is defined as proximity in distance measured in the domain or range space. The use of the term *neighborhood* is used loosely in this paper, as the elements in the *neighborhood* we define below are not necessarily immediate neighbors of the center pixel (in *topology*, it is commonly referred to as ϵ neighborhood). As before, a color image is represented as $f : X \to Y$, where X is a set of 2-D pixel positions and Y is a set of CIERGB tri-stimulus values.

Let E be a set of threshold or tolerance values. The neighborhood map $M_f : X \times E \to 2^X$ will be defined as a function from X and E to the set of all subsets of X. An important example of neighborhood maps is the domain ball set B (δ-neighborhood maps). Let $d_X : X \times X \to \mathbb{R}$ be a distance function in X and let $E = \mathbb{R}$. Using $x \in X$ and $\delta \in E$, define B as

$$B(x, \delta) = \{p \in X | d_X(x, p) \le \delta\}.$$

In other words, $B(x, \delta)$ is a set of points in X that are within δ distance from $x \in X$.

Similarly, neighborhood maps can be established using a distance function in the range space of f. With *a priori* knowledge that the end user of the output image is a human, pixels are discriminated using a distance metric in CIELAB space, which is normalized to human eye sensitivity. Let CIELAB color space be represented by the set $\hat{Y}$. The color space conversion map from Y to $\hat{Y}$ is denoted as $\pi : Y \to \hat{Y}$, $\pi([R, G, B]^T) = [L, a, b]^T$ where $[L, a, b]^T$ is a CIELAB value, as outlined in [3]. Let $y_1, y_2 \in Y$, $\hat{y}_1 = \pi(y_1)$, $\hat{y}_2 = \pi(y_2) \in \hat{Y}$, and

$$\begin{aligned} d_L(y_1, y_2) &= |\hat{y}_{1L} - \hat{y}_{2L}| \\ d_C(y_1, y_2) &= \sqrt{(\hat{y}_{1a} - \hat{y}_{2a})^2 + (\hat{y}_{1b} - \hat{y}_{2b})^2} \end{aligned} \tag{11}$$

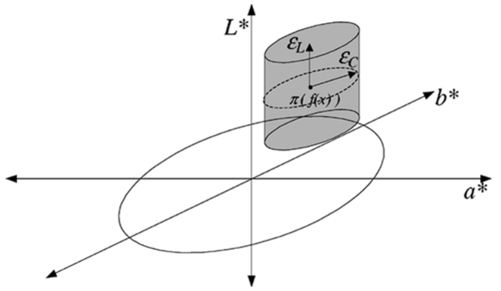
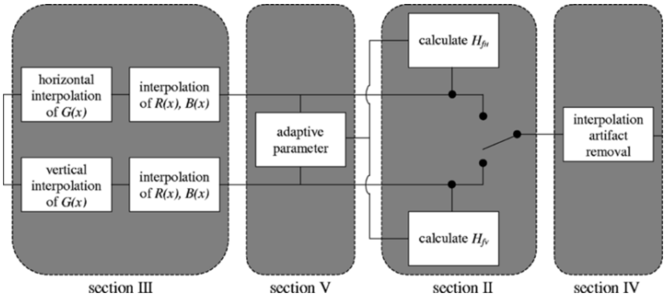


Fig. 8. Allowable range for $L_f(x, \epsilon_L) \cap C_f(x, \epsilon_C)$ in CIELAB space.



horizontal interpolation of G(x)
interpolation of R(x), B(x)
vertical interpolation of G(x)
interpolation of R(x), B(x)
adaptive parameter
calculate H_{f_H}
calculate H_{f_V}
interpolation artifact removal
section III
section V
section II
section IV

Fig. 9. Overview of the proposed algorithm

where $\hat{y}_i = [\hat{y}_{iL}, \hat{y}_{ia}, \hat{y}_{ib}]^T, i \in 1, 2$. In other words, d_L is a distance function using luminance (also known as ΔE_L), and d_C is a distance function in the a-b plane (also known as ΔE_{ab}) [14]. Using $x \in X$ and $\epsilon_L, \epsilon_C \in E = \mathbb{R}$, define a level neighborhood (level set) L_f and a color neighborhood (color set) C_f as follows:

$$\begin{aligned} L_f(x, \epsilon_L) &= \{p \in X | d_L(f(x), f(p)) \le \epsilon_L\} \\ C_f(x, \epsilon_C) &= \{p \in X | d_C(f(x), f(p)) \le \epsilon_C\}. \end{aligned} \tag{12}$$

L_f is a neighborhood established by distance in luminance (ΔE_L) and C_f is a neighborhood established by distance in color (ΔE_{ab}).

Define metric neighborhood $U_f : X \times E^3 \to 2^X (E = \mathbb{R})$ as

$$U_f(x, \delta, \epsilon_L, \epsilon_C) = B(x, \delta) \cap L_f(x, \epsilon_L) \cap C_f(x, \epsilon_C).$$

In other words, $U_f(x, \delta, \epsilon_L, \epsilon_C)$ is a set of all domain positions that are within δ distance of $x \in X$, within ϵ_L distance from luminance component of $\pi(f(x))$, and within ϵ_C distance from color component of $\pi(f(x))$. If $x_0 \in U_f(x, \cdot)$ then we expect that $f(x)$ appears similar to $f(x_0)$. Fig. 8 illustrates $L_f \cap C_f$, a subset of the range space $\hat{Y}$ which represents an allowable pixel values in U_f. When discrimination of pixels in luminance and chrominance are considered separately, $L_f \cap C_f$ forms a cylindrical shape in $\hat{Y}$; this is in contrast to the spherical shape, where luminance and chrominance are considered simultaneously to discriminate pixels by taking an Euclidean distance in $\hat{Y}$. This cylindrical shape is motivated by the reflectance model that describes how the spectral radiance is obtained as a product of the illuminant spectrum and the reflectance of the object at each wavelength. Whereas the variation in the luminance occur for

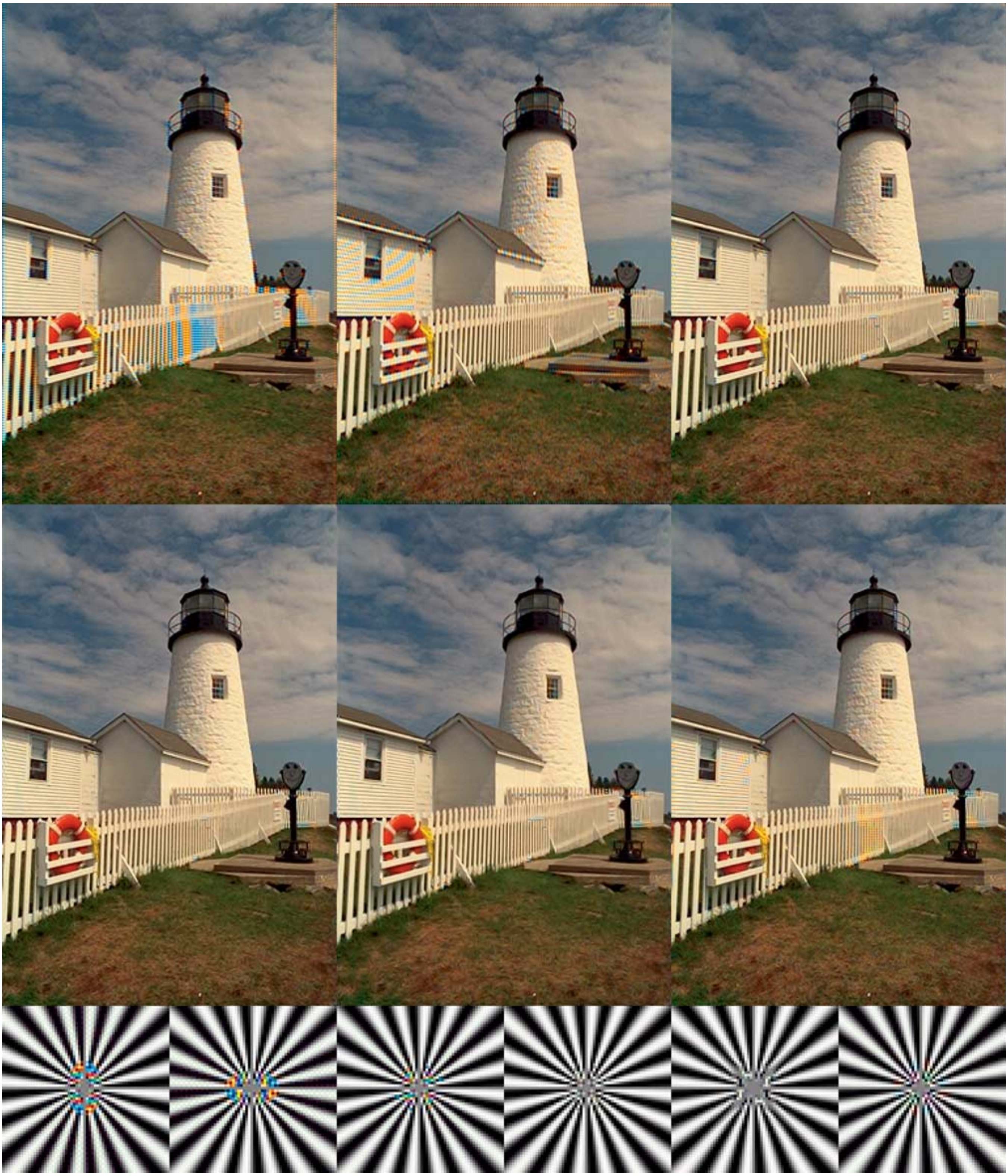


Fig. 10. Example outputs using “lighthouse” and “ray” images. From top to bottom, left to right: horizontal interpolation (f_H), vertical interpolation (f_V), directionality selection output (f), interpolation artifact removal (final output), and proposed method with forced $h_{0=0}$ (see text) method [8].

a number of reasons (e.g., surface angle), variation in chrominance is strongly tied with the object reflectance. Therefore, for the intended purpose of this paper, it is appropriate to consider luminance and chrominance separately. It was verified experimentally that the cylindrical shape approach is more effective for the proposed demosaicing algorithm.

Homogeneity is a tool designed to analyze the behavior of U_f. Define a homogeneity map $H_f : X \times E^3 \to \mathbb{R}$ as

$$H_f(x, \delta, \epsilon_L, \epsilon_C) = \frac{|U_f(x, \delta, \epsilon_L, \epsilon_C)|}{|B(x, \delta)|} \tag{13}$$

where $|\cdot| : 2^X \to \mathbb{R}$ is used to mean the size of the set.

B. Homogeneity and Artifacts

Misguidance artifacts occur when the direction of interpolation is *erroneously* selected. Interpolation along an object boundary is preferable to interpolation across this boundary because the discontinuity in the signal at the boundary contains high-frequency components that are difficult to estimate. When an image is interpolated in the direction orthogonal to the orientation of the object boundary, the color that appears at the pixel of interest is unrelated to the physical object represented in the image. Thus, a pixel marked by severe color artifacts has few pixels nearby that are similar, and we hypothesize that the misguidance color artifacts occur as isolated events. In other words, regions with color artifacts have smaller homogeneity map values (13).

Fig. 10 clearly illustrates this point. When the image containing vertical object boundaries are interpolated using a 2-D filter oriented in the horizontal direction, as in the white picket fence, several shades of orange and blue patches are seen on the white background. When the same image is interpolated using a 2-D filter oriented in the vertical direction, however, the image consists only of white and gray-colored surfaces. Fig. 11 corresponds to homogeneity maps for the lighthouse image in Fig. 10 (here, brighter white means larger homogeneity value). Homogeneity values corresponding to the vertical interpolation of the white picket fence are unambiguously larger than that of the horizontal interpolation.

With this simple observation, a homogeneity map can be used to compare the level of color artifacts present in multiple images. Suppose $\{f_1, f_2, \ldots f_P\}$ are color images, all reconstructed from the same CFA sampled data using different methods of interpolation. Let $\{H_1, H_2, \ldots, H_P\}$ be homogeneity maps corresponding to $\{f_1, f_2, \ldots f_P\}$, respectively. Then, at the pixel location $x \in X$, we argue that the pixel value $f_i(x)$ whose homogeneity value $H_i(x, \cdot)$ is the largest among $\{H_1, H_2, \ldots, H_P\}$ is least likely to have suffered from color artifacts.

In the demosaicing algorithm proposed in this paper, a homogeneity map is used to guide the direction of interpolation. More specifically, the interpolation procedure (Section II) takes the Bayer array data as its input, and produces horizontally interpolated (f_H) and vertically interpolated (f_V) color images. We are interested in combining f_H and f_V into a single image f by choosing to keep only the pixels interpolated in the direction with fewer artifacts. At each pixel location $x \in X$, only the pixel value whose corresponding homogeneity map value at x is larger is kept

$$f(x) = \begin{cases} f_V(x), & \text{if } H_{f_V}(x,\cdot) > H_{f_H}(x,\cdot) \\ f_H(x), & \text{if } H_{f_V}(x,\cdot) \leq H_{f_H}(x,\cdot). \end{cases} \tag{14}$$

It is easy to generalize this technique for combining more than two images; keep only the pixel value whose corresponding homogeneity map value is the largest among them.

Demosaicing algorithms with a directionality selection approach sometimes suffer from another type of artifact. Frequent switching from interpolation in one direction to another introduces discontinuities in the output images. For example, a thin, low-contrast line might be broken to pieces, especially if the input data is noisy. Human eyes are sensitive to patterns such as lines, so it is important to preserve them. Taking a spatial average of the homogeneity map reduces the discontinuity problem significantly. In other words, (14) is modified to

$$f(x) = \begin{cases} f_V(x), & \text{if } A * H_{f_V}(x,\cdot) > A * H_{f_H}(x,\cdot) \\ f_H(x), & \text{if } A * H_{f_V}(x,\cdot) \leq A * H_{f_H}(x,\cdot) \end{cases} \tag{15}$$

where $*$ denotes convolution, and A is a spatial averaging kernel, such as $A = [1,1,1;1,1,1;1,1,1]/9$.

The spatial averaging technique reduces misguidance color artifacts also. To see this, suppose $x_1 \in X$ is a pixel marked by a severe color artifact. It was noted earlier that x_1 has fewer pixels nearby that are similar, but an adjacent pixel $x_2 \in X$ will also have a low-homogeneity value because it is also unlikely that x_1 will be included in x_2's metric neighborhood.

IV. Interpolation Artifact Reduction

Even with a *perfect* directional selector, the output image from the interpolation algorithm [see (6) and (10)] may contain color artifacts. In this paper, this phenomenon is referred to as *interpolation artifacts*, and it is associated with limitations in the interpolation. Normally, interpolation artifacts are far less objectionable than misguidance artifacts (Section III), although they are still noticeable. Interpolation artifacts are reduced using a nonlinear technique similar to the iterative algorithms proposed by [8], [11], [13]. Recall that our hypothesis in Section II suggests that the difference images $R-G$ and $B-G$ are slowly varying. Let $\mathrm{median}(\cdot)$ denote a median filter operator. The following iterative procedure, performed m times, suppresses small variations in color while preserving edges

```
repeat_$m$_times
 1) $R = \mathrm{median}(R-G) + G$;
 2) $B = \mathrm{median}(B-G) + G$;
 1) $G = 1/2(\mathrm{median}(G-R) + \mathrm{median}(G-B) + R + B)$
end_repeat.
```

V. Complete Algorithm

A. Homogeneity-Directed Demosaicing Algorithm

The homogeneity-directed demosaicing algorithm is as follows.

```
1) Interpolate the mosaiced image in horizontal and vertical directions using (6) and (10). Let $f_H$ and $f_V$ stand for reconstructed color images from horizontal and vertical interpolation, respectively.
2) Evaluate homogeneity map and using (13).
3) Combine $f_H$ and $f_V$ into a single image, $f$, using (15).
4) Apply on $f$ the interpolation artifact reduction technique outlined in Section IV.
```

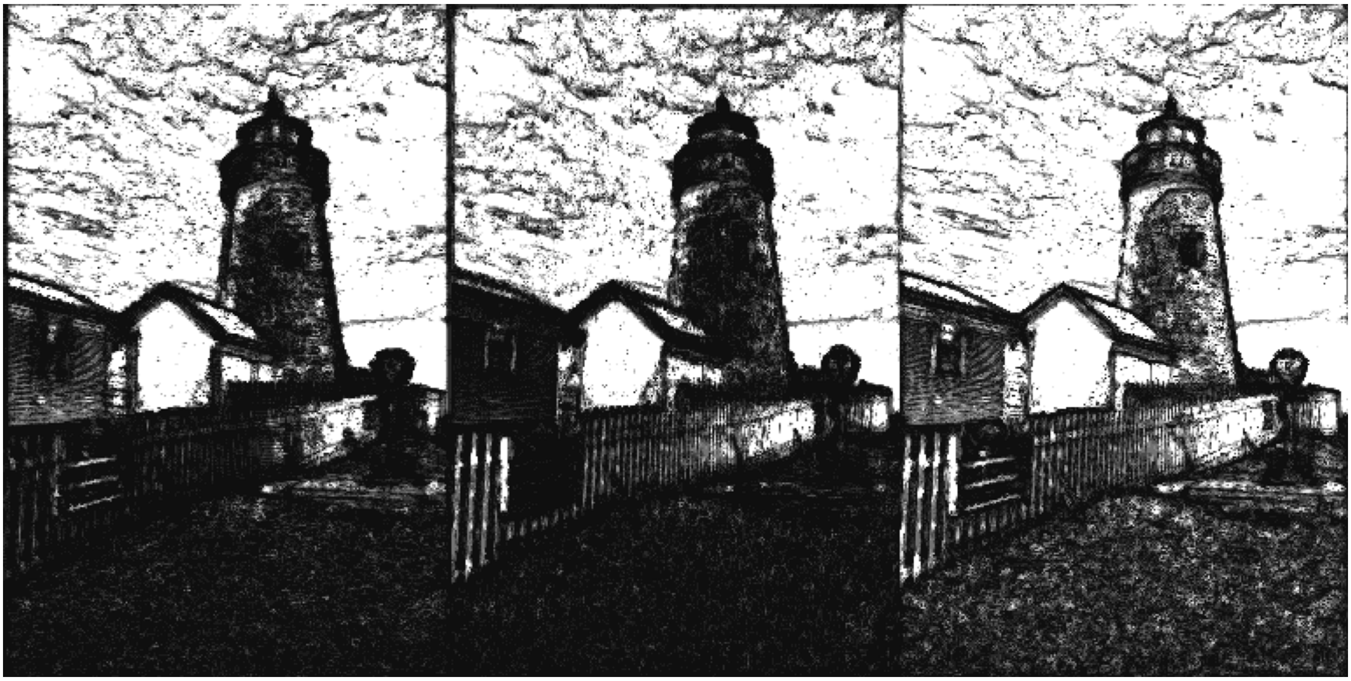


Fig. 11. Homogeneity maps corresponding to the lighthouse images in Fig. 10. From left to right: horizontally interpolated image, vertically interpolated image, and final output.

The authors did not consider interpolating in the diagonal directions. While filtering in directions other than horizontal and vertical may be an effective practice in general, incorporating diagonal interpolation in homogeneity-directed demosaicing algorithm takes considerable work. In Section III, we claimed that (15) minimizes the misguidance color artifacts. The effectiveness of (15) depends on the fact that when the direction of interpolation is *erroneously* selected, the color artifact is severe and the homogeneity values are small. This requires that the filtering coefficients have a high-pass component to it; indeed, h_0 filter in (6) is a horizontal (or vertical) bandpass filter. Due to the rectangular arrangements of the Bayer patterned CFA, it is difficult to design similar bandpass or high-pass filters in the diagonal directions. As a result, the low-passed diagonal filter will produce a smooth output image with high-homogeneity values, making (15) unfairly biased toward diagonal interpolation.

The "ray" image in Fig. 10 shows an experimental result verfying that horizontal and vertical filterbank interpolation is sufficient for handling diagonal edges. When $h_0 = 0$, however, the interpolated ray image suffers from severe artifacts except in the regions where the edges are *exactly* horizontal or vertical. We may, therefore, conclude that the filter bank interpolation scheme (6) is responsible for how well diagonal edges are interpolated.

B. Adaptive Parametrization

The homogeneity map $H_f(x, \cdot)$ used in step 2 above needs parameters δ, ϵ_L, and ϵ_C. The domain tolerance δ controls the computational complexity of the algorithm. In this paper, δ is fixed. If ϵ_L and ϵ_C are fixed values also, the algorithm performs poorly on low-contrast edges because the metric neighborhood model is unable to distinguish between the pixels that belong to low-contrast physical objects.

To overcome this difficulty, the parameter information is extracted from the scene adaptively. For each pixel location $x \in X$, we are interested in assigning ϵ_L and ϵ_C to values that reflect *typical* variations among pixels belonging to the same object [see (12)].

Assume for a moment that the orientation of the object boundary at $x \in X$ is horizontal. Thus, we want $f(x) = f_H(x)$. In this case, the pixels located immediately to the right and the left of x are likely to belong to the metric neighborhood $U_f(x, \cdot)$. Thus, a good candidate for the parameter values ϵ_L and ϵ_C at x is the larger of the distances between $f_H(x)$ and each of the adjacent pixels immediately to the right and to the left. That is, let $x = [x_H, x_V]^T$, where x_H is the horizontal coordinate, and x_V is the vertical coordinate. Then

$$\begin{aligned} \epsilon_{LH}(x) &= \max_{x_0 \in \{x - x_1, x - x_2\}} d_L(f_H(x), f_H(x_0)) \\ \epsilon_{CH}(x) &= \max_{x_0 \in \{x - x_1, x - x_2\}} d_C(f_H(x), f_H(x_0)) \end{aligned} \tag{16}$$

where $x_1 = [1, 0]^T$, $x_2 = [-1, 0]^T$. Likewise, if the orientation of the object boundary at x is vertical instead, then consider the larger of the distances between $f_V(x)$ and the adjacent pixels immediately above and immediately below. That is

$$\begin{aligned} \epsilon_{LV}(x) &= \max_{x_0 \in \{x - x_3, x - x_4\}} d_L(f_V(x), f_V(x_0)) \\ \epsilon_{CV}(x) &= \max_{x_0 \in \{x - x_3, x - x_4\}} d_C(f_V(x), f_V(x_0)) \end{aligned} \tag{17}$$

where $x_3 = [0, 1]^T$, $x_4 = [0, -1]^T$.

To assign ϵ_L and ϵ_C to values that reflect *typical* variations among pixels belonging to the same object, we propose the following scheme:

$$\begin{aligned} \epsilon_L(x) &= \min\{\epsilon_{LH}(x), \epsilon_{LV}(x)\} \\ \epsilon_C(x) &= \min\{\epsilon_{CH}(x), \epsilon_{CV}(x)\}. \end{aligned} \tag{18}$$

TABLE I
AVERAGE S-CIELAB ERROR MAGNITUDE

		method	proposed	proposed		fixed				method	proposed	proposed		fixed	
image	bilinear	in	method	method	RG B	paramter	l^1	image	bilinear	in	method	method	RGB	parameter	l^1
no		[8]	$\delta = 2$	$\delta = 1$	space	$\epsilon = 1$	metric	no		[8]	$\delta = 2$	$\delta = 1$	space	$\epsilon = 1$	metric
1	11. 06	5. 32	4. 67	4. 63	4.80	5.26	4.63	11	14. 42	8. 06	7. 79	7. 82	8.03	9.03	7.87
2	13. 63	9. 36	7.13	6. 99	7.11	7.77	7.12	12	6.94	5.46	4.49	4.36	4.45	4.54	4.37
3	22.35	13.09	10.23	10.23	10.24	12.24	10.29	13	6.43	3.48	2.44	2.34	2.46	2.54	2.34
4	7. 96	4.28	3.07	2.93	3.13	3.29	2.93	14	7.90	5.28	4.34	4.27	4.27	4.48	4.32
5	6.22	4.63	3.72	3.53	3.50	3.95	3.60	15	16.94	11.13	10.19	10.12	10. 02	11.13	10.17
6	13.18	5.94	5.02	4. 95	5.08	5.61	4.98	16	9.42	4.44	4.51	4.44	4.46	4.87	4.50
7	8.30	5.57	4.01	3.95	3.91	4.77	3.99	17	7.37	5.11	4.47	4.17	4.14	4. 69	4.15
8	4.28	2.93	2.21	2.10	2.14	2.19	2.10	18	7.57	4.39	3.85	378	3.89	4.34	3.79
9	8.45	4.89	3.94	3.87	4.00	4.28	3.92	19	6.37	4.58	4.31	4.20	4.22	4.43	4.26
10	3.82	3.01	1.96	1.67	1.72	1.73	1.67	20	7.68	4.67	4.42	4.40	4.32	4.68	4.38

The homogeneity maps $H_{f_H}(x,\cdot)$ and $H_{f_V}(x,\cdot)$ are computed using the parameters (18).

As discussed in Section IV, f_H and f_V may have interpolation artifacts, especially in textured and edge regions. Thus in some cases, ϵ_L and ϵ_C are affected by interpolation artifacts. The purpose of using metric neighborhood modeling, however, is to discriminate misguidance artifacts. Because the misguidance artifacts are significantly more severe than interpolation artifacts, ϵ_L and ϵ_C are sufficient enough to exclude misguidance artifacts from the metric neighborhood.

Fig. 9 shows an overview of the proposed algorithm with the adaptive parameter selection.

VI. IMPLEMENTATION AND RESULTS

A. Results

In this section, the adaptive homogeneity-directed demosaicing algorithm is implemented using $\delta = 2$. A bilinear interpolator is used for L [see (10)]. The filter coefficients for the filter $h(x)$ are chosen by approximating (7) with powers of 2 to reduce computational costs

$$h = \frac{[-1, 2, 2, 2, -1]}{4}.$$

Interestingly, the approximated filter coefficients are identical to the filter used in [1]. The interpolation artifact reduction step (see Section IV) is iterated three times—this is determined empirically.

The "lighthouse" and "ray" images, sampled using Bayer pattern CFA, are used as the test input images in Fig. 10. Images interpolated using 2-D filters oriented horizontally (f_H) and vertically (f_V) show zippering artifacts [see (6) and (10)]. Fig. 12 shows the directionality selection procedure (step 3, Section V-A) in detail. When the algorithm selects the vertical direction to interpolate, it is marked by a white pixel. If horizontal interpolation has the larger homogeneity value instead, then it is marked by a black pixel. It is clear that our metric neighborhood is modeling misguidance color artifacts well from the output images from (15). The iterative interpolation artifact reduction step reduces the purple and green colors smearing out the diagonal roofline, makes the lighthouse rails more black, and whitens the parts of the fence that appear blue. The reconstructed "ray" image, synthetically generated to mimic a resolution chart, demonstrates that horizontal and vertical interpolations are adequate for interpolating diagonal lines.

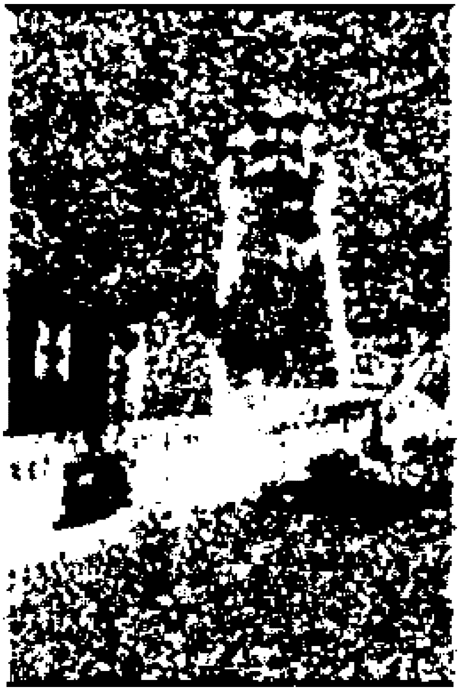


Fig. 12. Directionality selection for lighthouse image in Fig. 10: white pixel if $H_{f_V} < H_{f_H}$; black if otherwise.

Table I compares the proposed algorithm (with $\delta = 2$) with bilinear interpolation and a method developed by Gunturk *et al.* [8] using a human perceptual error measurement known as S-CIELAB space [17]. The table shows the average S-CIELAB error magnitudes of output images, and the 20 test images used here are same as the images used in [8]. The table clearly shows a significant gain over bilinear interpolation and an improvement over [8]. A careful visual inspection of the output images reveals important differences between the method in [8]

TABLE II
COMPUTATIONS PERFORMED PER 1 PIXEL OUTPUT ($\delta = 2$)

	adds	fixed multiplies	multiplies	compares	LUTs	absolute values
Interpolation	15	5	0	0	0	0
Color Space Conversion	16	18	0	0	6	0
Homogeneity	48	0	24	49	0	12
Adaptive Parameters	0	0	0	6	0	0
Median Filtering (per iteration)	9	1	0	16	0	0

and the proposed algorithm. In some cases, the proposed algorithm smoothes out the very small details in the image. But the proposed algorithm reduces the level of color artifacts when compared to [8], and does not suffer from zippering artifacts (see fence in Fig.10). The proposed algorithm shows its strength especially when interpolating repetitive patterns. These trends come as no surprise, since the proposed algorithm is optimized for color consistency, while [8] is optimized for geometry. We encourage reviewing [9] to see more output images from the proposed algorithm and to compare them to the results from other demosaicing algorithms.

B. Computational Complexity

The computational complexity of this algorithm (parameters and filters as outlined in Section V-A) is summarized below. Calculations performed for each pixel are listed in Table II. *Fixed* multiplication means multiplication by a fixed coefficient, which can be hard coded with cumulative shift adds. Furthermore, the multiplications that appear in interpolation and median filtering steps are powers of two; they may be substituted with bit shifts.

In many realtime applications today, it is appropriate to consider tradeoffs between image quality and complexity by choosing suboptimal designs. Below, we examined four suboptimal design choices, and the tradeoffs are numerically presented in terms of S-CIELAB error magnitude in Table I.

First, reducing δ, the radius of the domain ball set is the most straight-forward way to reduce complexity of the algorithm. For example, although the metric neighborhood is made more local when $\delta = 1$, the quality of the output images is comparable to and often better than that of $\delta = 2$ (see Table I, $\delta = 1$). When $\delta = 1$, the computational complexity of homogeneity reduces drastically (16 adds, four absolute values, eight multiplies, and 17 compares).

Second, color space conversion into CIELAB color space is expensive because it is necessary to exercise 6 LUTs and 18 multiplications. Use of suboptimal color spaces such as Yc_bc_r (12 adds, 18 multiplies, and 0 LUTs) and RGB (0 adds, 0 multiplies, and 0 LUTs) are considered. The output images based on Yc_bc_r homogeneity maps yield reasonable image quality, and images from RGB homogeneity maps may also be acceptable. The S-CIELAB error magnitude for RGB homogeneity-directed demosaicing algorithm is shown in Table I.

Third, fixed parametrization is considered, replacing (18) with fixed numbers $\epsilon_L = 1$, $\epsilon_C = 1$. The performance of the fixed parametrization is poor, and the output images are usually unacceptable under visual inspection. We conclude that the effectiveness of the proposed algorithm is due largely to adaptive parametrization (see Table I, fixed parameter).

Finally, use of suboptimal distance metric is considered. The homogeneity maps (13) assume a ℓ^2 norm distance function (11), which may be substituted with a ℓ^1 norm distance metric. This interchange would eliminate 24 multiplications involved in calculating the homogeneity maps. The output image quality using ℓ^1 norm is comparable to that of ℓ^2 norm (see Table I) ℓ^1 distance metric.

Memory efficiency plays an important part of the design if the algorithm is implemented as an ASIC pipeline. Suppose the pixels are processed in raster scan order. Then, an increase in $h(x)$ filter length by one means an additional pixel buffering by one raster line. Likewise, this paper only considers using only one row (column) of data for interpolation of green pixels. By considering multiple rows (columns), the memory bandwidths grow quickly.

VII. CONCLUSION

In this paper, an adaptive homogeneity-directed demosaicing algorithm was presented. Metric neighborhood modeling techniques were used to compare the level of color artifacts that are present in images and to select the direction for interpolation. Filterbank interpolation techniques were developed to cancel aliasing, and interpolation artifact reduction iterations suppressed color artifacts. Experimental data shows how well the algorithm performs.

ACKNOWLEDGMENT

The authors would like to thank Texas Instruments and Agilent Technologies for helpful discussions and encouragement. They would also like to thank B. K. Gunturk for providing some test images and the code to his algorithm.

REFERENCES

[1] J. E. Adams Jr., "Design of color filter array interpolation algorithms for digital cameras, Part 2," in *IEEE Proc. Int. Conf. Image Processing*, vol. 1, Oct. 1998, pp. 488–492.

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

[3] "Colorimetry," Central Bureau CIE, Vienna, Austria, CIE Pub. no. 15.2.

[4] D. R. Cok, "Reconstruction of CCD images using template matching," in *Proc. IS&T Annu. Conf.*, vol. 2, May 1994, pp. 380–385.

[5] H. Figueroa, "Density based image analysis," in *Proc. Soc. Industrial and Applied Mathematics Annu. Meeting*, Jul. 2000.

[6] W. T. Freeman and Polaroid Corporation, "Method and apparatus for reconstructing missing color samples," U.S. Patent 4 774 565, 1998.

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

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

[9] B. K. Gunturk, J. Glotzbach, Y. Altunbasak, R. W. Schafer, and R. M. Mersereau, "Demosaicking: Color filter array interpolation in single chip digital cameras," *IEEE Signal Process. Mag.*, no. 9, Sep. 2004.

[10] K. Hirakawa and T. W. Parks, "Adaptive homogeneity-directed demosaicing algorithm," in *Proc. IEEE Int. Conf. Image Processing*, vol. 3, Sep. 2003, pp. 669–672.

[11] R. Kimmel, "Demosaicing: Image reconstruction from CCD samples," *IEEE Trans. Image Process.*, vol. 8, no. 6, pp. 1221–1228, Jun. 1999.

[12] 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.

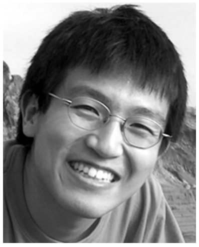
[13] D. D. Muresan and T. W. Parks, "Optimal recovery demosaicing," in *Proc. IASTED Signal and Image Processing*, Aug. 2002, pp. 260–265.

[14] G. Sharma and H. J. Trussell, "Digital color imaging," *IEEE Trans. Image Processing*, vol. 6, no. 7, pp. 901–932, Jul. 1997.

[15] W. Sweldens, "The lifting scheme: A new philosophy in biorthogonal wavelet constructions," *Proc. SPIE*, pp. 68–79, 1995.

[16] G. Wyszecki and W. S. Stiles, *Color Science: Concepts and Methods, Quantitative Data and Formulae*. New York: Wiley, 1982.

[17] X. Zhang, D. A. Silverstein, J. E. Farrell, and B. A. Wandell, "Color image quality metric S-CIELAB and its application on halftone texture visibility," in *Proc. IEEE COMPCON*, Feb. 1997, pp. 44–48.



Keigo Hirakawa (S'04) received the B.S. degree in electrical engineering from Princeton University, Princeton, NJ, in 2000 and the M.S. degree in electrical and computer engineering from Cornell University, Ithaca, NY, in 2003. He is currently pursuing the Ph.D. degree at Cornell University.

His research interests include image modeling, color representation, image denoising, and image interpolation.

Thomas W. Parks (S'66–M'67–SM'79–F'82) received the Ph.D. degree from Cornell University, Ithaca, NY, in 1967.

From 1967 until 1986, he was with the Department of Electrical and Computer Engineering, Rice University, Houston, TX. In 1986, he joined Cornell University as a Professor of electrical engineering. He has coauthored several books on digital signal processing and published a number of research papers. His research interests are signal theory and digital signal processing.

Prof. Parks is a recipient of the IEEE Third Millennium Medal and the 2004 IEEE Jack S. Kilby Medal. He received the Humboldt Foundation Senior Scientist Award and has been a Senior Fulbright Fellow.