Stochastic Color Interpolation for Digital Cameras

Hung-An Chang and Homer H. Chen, *Fellow, IEEE*

***Abstract*—This paper presents a stochastic estimation approach to adaptive interpolation of color filter array. It models an image as a 2-D locally stationary Gaussian process and achieves robustness against aliasing by employing an edge-sensitive weighting policy based on the stochastic characteristics of uniformly oriented edge indicators. Experimental results show that the algorithm can effectively eliminate the occurrence of perceptible artifact. Performance comparison in terms of peak signal-to-noise ratio and mean square error is provided to demonstrate the superiority of the proposed algorithm.**

***Index Terms*—Color filter array (CFA), color interpolation, demosaicing, edge detection, edge-sensitive weighting.**

I. Introduction

MOST digital cameras with a single CCD or CMOS sensor array use a color filter array (CFA) to capture images. The most commonly used CFA is the Bayer color array, or Bayer pattern [1], that samples the blue and green colors alternatively in the even rows and the green and red colors in the odd rows as shown in Fig. 1, resulting in a subsampled image with a single color component for each pixel. The green pixels are sampled at a higher rate because human eye is more sensitive to the green color than to the blue and red colors. To reconstruct a full color image from the raw data, color interpolation (also known as demosaicking) is required to estimate the two missing color components of each pixel.

The most straightforward color interpolation algorithm is bilinear interpolation, which determines the missing colors of each pixel by averaging the color values of nearby pixels. However, it suffers from artifacts such as false color and blurring especially in image areas with dense edges. This type of artifacts is also known as the zipper effect and is caused by alias—a situation where the Nyquist rates of the high spatial frequency bands of such image areas exceed the sampling rate of the CFA. As illustrated in Fig. 2, the artifacts become more profound for sharper edges because the energy of the high frequency bands increases as the edges become sharper.

Many techniques [2]–[24] have been proposed to mitigate the alias effect. One class of techniques exploits the high correlation between different color channels and transforms the image to a new color space where the image intensity field varies more slowly. For example, Cok [2] found that the hue of pixels, namely R/G or B/G, has a slower variation than *RGB* and proposed a color interpolation method over hues. Pei *et al.* [4] found that the color differences G-R and G-B also exhibit slowly varying characteristics and proposed a color difference based interpolation method. However, these methods are unable to handle images where the color values of different channels do not change coherently and hence sharp edge may remain after the color space transformation.

Another important class of aliasing reduction techniques tries to solve the problem by edge-adaptive interpolation. The main idea is to adjust the weight for interpolation by using edge indicators along various edge directions. An edge indicator along an edge direction estimates the likelihood of edge crossing when moving from the pixel to be interpolated to another pixel in the

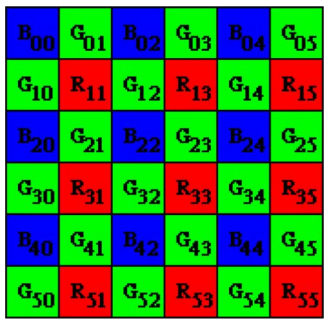

Fig. 1. Bayer color array.

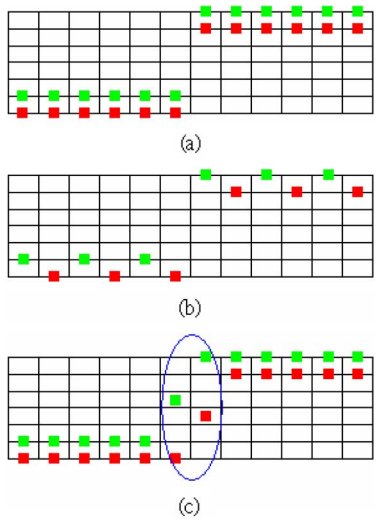

Fig. 2. Illustration of the zipper effect along a scan line of an image. Only green and red pixels are shown (modified from [6]). Vertical axis represents the color intensity. (a) Original image. (b) Bayer pattern image. (c) Image obtained by bilinear interpolation suffers from the zipper effect near the edge.

Manuscript received November 6, 2006; revised February 10, 2007. This work was supported in part by Avisonic Technology Company, Hsin-Chu, Taiwan, R.O.C. This paper was recommended by Associate Editor S. Takamura.

H.-A. Chang was with the Department of Electrical Engineering, National Taiwan University, Taipei 10617, Taiwan, R.O.C. He is now with the Computer Science and Artificial Intelligence Laboratory, Massachusetts Institute of Technology, Cambridge, MA 02142 USA.

H. H. Chen is with the Department of Electrical Engineering, Graduate Institute of Communication Engineering, and Graduate Institute of Networking and Multimedia, National Taiwan University, Taipei 10617, Taiwan, R.O.C. (e-mail: homer@cc.ee.ntu.edu.tw).

Color versions of some or all figures available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TCSVT.2007.897471

edge direction. The weight for interpolation is inversely proportional to the likelihood estimate to prevent interpolating across an edge.

Gradient-based interpolation [20] is a simple version of edge-adaptive interpolation. It uses the vertical and horizontal gradients as the edge indicators and avoids pixels in the direction of large gradient in the interpolation. This method can be regarded as a two-directional, hard-decision edge adaptive method. The edge-adaptive interpolation can be further refined by taking more edge directions into account and by applying a soft-decision weighting policy [22]. The C2D2 method proposed by Kêhtarnavaz *et al.* [5] considers eight directions and uses the reciprocal of the directional derivatives as the weights for interpolation and achieves significant improvement over the gradient method.

The success of the C2D2 method suggests that the performance of edge-adaptive interpolation depends on its edge sensing ability and the robustness of its adaptive weighting policy. Motivated by this observation, we develop a stochastic approach to color interpolation. Specifically, the image is modeled as a 2-D locally stationary Gaussian process and the interpolation as stochastic estimation. Each missing color of a pixel is interpolated from a set of uniformly oriented candidate pixels based on the edge indicators and the observable data of the candidates. The relative weight of the estimate for each candidate is stochastically determined according to a reliability measure that is computed based on the statistics of the corresponding edge indicator. Then, the missing color is obtained by computing the weighted average of the estimates. Experimental results show that the algorithm proposed here can effectively eliminate the occurrence of perceptible artifact experienced in previous methods.

The rest of the paper is organized as follows. In Section II, the properties of a locally stationary Gaussian process are described. The details of the proposed algorithm, including the calculation of edge indicators and the stochastic weighting policy, are described in Section III. Experimental results and discussions are given in Section IV, followed by a summary of the paper in Section V.

II. Locally Stationary Gaussian Process

We begin this section by introducing the autocorrelation function of a locally stationary Gaussian process and expressing the difference of two samples in terms of the autocorrelation function. Then, we show how to normalize the differences of samples to so that they can be mapped into an identical Gaussian distribution for further probabilistic analysis.

Within a locally stationary region, the autocorrelation function $\Omega(d)$ of an image intensity field [3] can be expressed as

$$\Omega(d) = \exp^{(-d^2/\rho^2)} \tag{1}$$

where d is the distance between the two pixels considered for the function, and ρ is an image-dependent parameter whose value can change from one locally stationary region to another within an image. This model implies that the power spectral density of the intensity field within a locally stationary region is also a Gaussian distribution and in accordance with the fact that the high frequency bands possess much less energy than the low frequency bands. Such high frequency bands occupy only a small portion of the image spectrum, but they are the one that causes the notorious perceptible aliasing.

For two samples of the process, $X(t_1)$ and $X(t_2)$, belonging to a locally stationary region and with $|t_1 - t_2| = d$, the difference $X(t_1) - X(t_2)$ is a zero-mean Gaussian random valuable. Because of the stationary property, the standard deviation of $X(t_1) - X(t_2)$ is the same as that of $X(d) - X(0)$. Denote the standard deviation by σ_d; we have (2), shown at the bottom of the page, where μ_X and σ_X, respectively, are the mean and standard deviation of the process. For convenience, we denote $X(d) - X(0)$ by $\mathrm{Diff}(d)$.

Consider a set of N differences $\{\mathrm{Diff}(d_n), 1 \leqslant n \leqslant N\}$. All these differences are Gaussian with zero mean, but depending on d_n their standard deviations may be different. By setting a unit distance d and multiplying $\mathrm{Diff}(d_n)$ by a normalization factor κ_n as

$$\begin{aligned}\kappa_n &= \frac{\sigma_X\sqrt{2(1-\Omega(d))}}{\sigma_X\sqrt{2(1-\Omega(d_n))}} = \frac{\sqrt{(1-\Omega(d))}}{\sqrt{(1-\Omega(d_n))}} \\ &= \frac{\sqrt{(1-\Omega(d))}}{\sqrt{(1-(\Omega(d))^{d_n^2/d^2})}}\end{aligned} \tag{3}$$

where the autocorrelation function expressed in (1) is used in the derivation, we can map $\mathrm{Diff}(d_n)$ into a Gaussian distribution with zero mean and standard deviation σ_d.

Note that the normalization factor κ_n is a slowly varying number when the value of $\Omega(d)$ is close to 1. To see this, let us consider the case where $d = 2$, and $d_n = 2\sqrt{5}$. As $\Omega(d)$ varies from 0.85 to 0.95, κ_n changes from 0.519 to 0.472, representing only a 7% change. In this case, κ_n can be approximated by a constant 0.5 to simplify the computation.

III. Stochastic Color Interpolation

The proposed interpolation algorithm is divided into three steps as shown in Fig. 3. In the first step, the missing green components of blue and red pixels are interpolated. Next, the missing red components of blue pixels and the missing blue components of red pixels are interpolated. Finally, the missing blue and red components of green pixels are interpolated. The missing green components are interpolated first because the G channel has more samples available in the Bayer pattern and

$$\begin{aligned}\sigma_d &= \sqrt{E[(X(d)-X(0))^2]} \\ &= \sqrt{E[((X(d)-\mu_X)-(X(0)-\mu_X))^2]} \\ &= \sqrt{E[(X(d)-\mu_X)^2]+E[(X(0)-\mu_X)^2]-2E[(X(d)-\mu_X)(X(0)-\mu_X)]} \\ &= \sigma_X\sqrt{2(1-\Omega(d))},\end{aligned} \tag{2}$$

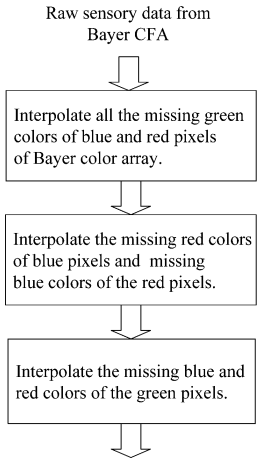

Fig. 3. Flowchart of the stochastic color interpolation algorithm.

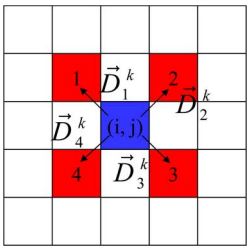

Fig. 4. Example of interpolation with four condidate pixels.

hence can generate more reliable estimate than the other two channels. A reliable estimation of the green components also helps enhance the interpolation of red and blue components because of the high correlation between these channels. In this algorithm we separate the interpolation of R and from that of B because the numbers of candidate pixels for these two components are different.

In this section, we describe the edge indicators and the stochastic weighting policy and show the interpolation formula used in each step of the algorithm. The index k refers to the step number (which can be 1, 2, or 3) in the following discussion.

A. Edge Indicators

Suppose we want to interpolate a missing color S (representing R, G, or B) of the pixel located at (i,j) (meaning ith row and jth column) of the Bayer pattern from N_k candidate pixels. Denote the set of displacements (in terms of pixels) of these candidates with respect to (i,j) by $\{\vec{D}^k_n = (v^k_n, h^k_n), 1 \leqq n \leqq N_k\}$, where v^k_n is the vertical displacement from (i,j) to the nth candidate pixel and h^k_n the horizontal displacement. Fig. 4 shows an example of these notations for the interpolation of red color $(S = R)$. In this example, $N_k = 4, (v^k_1, h^k_1) = (-1,-1)$, $(v^k_2, h^k_2) = (-1,1)$, $(v^k_3, h^k_3) = (1,1)$, and $(v^k_4, h^k_4) = (1,-1)$.

The candidate pixels considered in the interpolation algorithm have the following properties.

1) They have the S color component obtained from either the original CFA or the previous steps.
2) They are symmetric. That is, if the pixel at $(i+v^k_n, j+h^k_n)$ is a candidate, the pixel at $(i-v^k_n, j-h^k_n)$ is also a candidate.
3) For each n, the pixel at $(i+2v^k_n, j+2h^k_n)$ has the same color as the pixel at (i,j).

Define the unit distance $d = 2\sqrt{(v^k_1)^2 + (h^k_1)^2}$ and the nth edge indicator $E^k_n(i,j)$ as

$$E^k_n(i,j) = 0.5\kappa_n \left(\left|C(i+v^k_n, j+h^k_n) - C\left(i-v^k_n, j-h^k_n\right)\right| + \left|C\left(i+2v^k_n, j+2h^k_n\right) - C(i,j)\right|\right) \quad (4)$$

where $C(i,j)$ denotes the value of CFA sensory data at (i,j), and κ_n is the value obtained from (3) with $d_n = 2\sqrt{(v^k_n)^2 + (h^k_n)^2}$. Given d and d_n, the value of κ_n as mentioned in Section II can be approximated by a constant to save computation. Then the distributions of $E^k_n(i,j)$ for all n and k become the same.

Note that $C(i,j)$ and C$(i + v^k_n, j + h^k_n)$ in (4) are of different colors. The reason that we consider two color channels for the edge indicator is to increase the edge sensitivity. Because a sharp change in any color channel can result in a sharp edge, jointly considering two color channels can improve the edge-detecting ability.

B. Stochastic Weighting Policy

In step k, the missing color $x^k(i,j)$ is estimated by

$$\hat{x}^k(i,j) = \sum_{n=1}^{N_k} x^k_n(i,j) w^k_n(i,j) \quad (5)$$

where $x^k_n(i,j)$ is an estimate of $x^k(i,j)$ based on the data available at $(i+v^k_n, j+h^k_n)$, and $w^k_n(i,j)$ is the weight associated with $x^k_n(i,j)$.

The weight $w^k_n(i,j)$ is computed according to the reliability of $x^k_n(i,j)$, which is the probability of $x^k_n(i,j)$ being a reliable estimate. Conceptually, we can consider that the reliability of $x^k_n(i,j)$ is measured with respect to a reference edge at the current pixel (i,j). Denote the reference edge by e and its magnitude by $|Z|$, which is the absolute difference of two pixels separated by one unit distance across the edge. If the value of the edge indicator $E^k_n(i,j)$ is larger than $|Z|$, it implies that there is an edge sharper than e between the current pixel (i,j) and the candidate pixel $(i+v^k_n, j+h^k_n)$. Therefore, the estimate $x^k_n(i,j)$ is unreliable. On the other hand, if $E^k_n(i,j) < |Z|, x^k_n(i,j)$ is a potentially reliable estimate. With this concept in mind and under the stochastic framework discussed so far, we treat $|Z|$ as a random variable. We compute the probability $P(|Z| > E^k_n(i,j))$ and use it as the reliability of the estimate $x^k_n(i,j)$. The weight $w^k_n(i,j)$ for $x^k_n(i,j)$ is assigned in proportion to the reliability of $x^k_n(i,j)$; hence it is proportional to $P(|Z| > E^k_n(i,j))$. The task now is to estimate the distribution of $|Z|$ based on the information we have, namely, the values of the edge-indicators around (i,j).

With the image modeled as a locally stationary Gaussian process, Z becomes a zero-mean Gaussian with standard deviation σ_d. Therefore, the cumulative distribution function (cdf) of $|Z|$ is

$$P(|Z| \leqslant t) = \int_0^t \frac{2}{\sqrt{2\pi}\sigma_d} e^{\frac{-z^2}{2\sigma_d^2}}\, dz. \quad (6)$$

The weight $w_n^k(i,j)$ for $x_n^k(i,j)$ is proportional to the reliability of $x_n^k(i,j)$ and thus satisfies the following two conditions:

$$w_n^k(i,j) \propto P(|Z| > E_n^k(i,j)) = \int_{E_n^k(i,j)}^{\infty} \frac{2}{\sqrt{2\pi}\sigma_d} e^{\frac{-z^2}{2\sigma_d^2}}\, dz \tag{7}$$

$$\sum_{n=1}^{N_k} w_n^k(i,j) = 1. \tag{8}$$

To calculate the weight, we need to find the value of σ_d first. Instead of calculating σ_d directly, we can find its value from the expectance of $|Z|$ because

$$\begin{aligned} \mathrm{Exp}[|Z|] &= \int_0^{\infty} z \frac{2}{\sqrt{2\pi}\sigma_d} e^{\frac{-z^2}{2\sigma_d^2}}\, dz \\ &= \frac{2}{\sqrt{2\pi}\sigma_d}\left(\sigma_d^2\right)\left(-\, e^{\frac{-z^2}{2\sigma_d^2}}\Big|_0^{\infty}\right) \\ &= \sqrt{\frac{2}{\pi}}\sigma_d \end{aligned} \tag{9}$$

which is proportional to σ_d. Once the expectance of $|Z|$ is known, σ_d can be determined immediately. The expectance of $|Z|$ can be easily calculated from the edge indicators $E_n^k(i,j)$. Due to the locally stationary property of the image model, the difference values $\kappa_n|C(i+v_n^k, j+h_n^k) - C(i-v_n^k, j-h_n^k)|$ and $\kappa_n|C(i,j) - C(i+2v_n^k, j+2h_n^k)|$ used for calculating the edge indicators defined in (4) have the same distribution as $|Z|$; thus the expectance of the average of these difference values equals the expectance of $|Z|$. That is

$$\mathrm{Exp}\left[\frac{1}{N_k}\sum_{n=1}^{N_k} E_n^k(i,j)\right] = \mathrm{Exp}[|Z|] = \sqrt{\frac{2}{\pi}}\sigma_d. \tag{10}$$

Since the average is an unbiased estimate of the expectance, the value of σ_d is estimated by averaging the edge indicator directly.

Because the integration in (7) is not in closed form, we precompute the weights and store them in a look-up table to achieve computational efficiency. Given a specified step size δy, we calculate the numerical value of

$$f(y_m) = 2(1 - \Phi(y_m)) = \int_{y_m}^{\infty} \frac{2}{\sqrt{2\pi}} e^{-\frac{u^2}{2}}\, du \tag{11}$$

where $y_m = m \cdot \delta y$ and $\Phi(y_m)$ is the cdf of standard Gaussian distribution, and store it in the mth entry of the table. The probability $P(|Z| > E_n^k(i,j))$ in (7) can be expressed as

$$\begin{aligned} P(|Z| > E_n^k(i,j)) &= \int_{E_n^k(i,j)}^{\infty} \frac{2}{\sqrt{2\pi}\sigma_d} e^{-\frac{z^2}{2\sigma_d^2}}\, dz \\ &= \int_{E_n^k(i,j)/\sigma_d}^{\infty} \frac{2}{\sqrt{2\pi}} e^{-\frac{u^2}{2}}\, du \\ &= f\left(E_n^k(i,j)/\sigma_d\right) \\ &= f\left(\sqrt{2/\pi}E_n^k(i,j)/\mu\right) \end{aligned} \tag{12}$$

where μ is the average of the edge indicators

$$\mu = \sum_{n=1}^{N_k} E_n^k(i,j)/N_k. \tag{13}$$

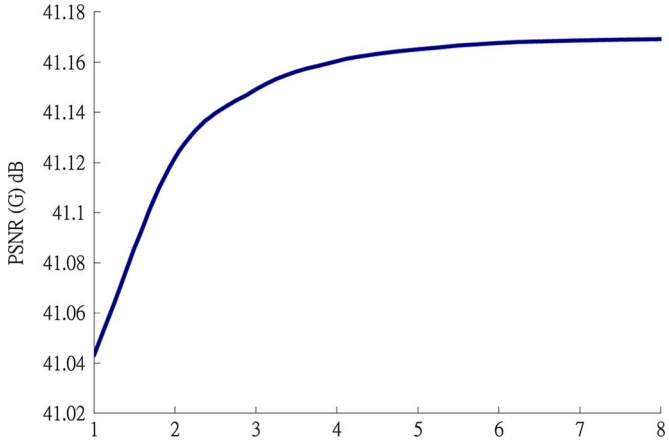

Fig. 5. Performance of the algorithm under different step sizes.

TABLE I
LOOK-UP TABLE OF $f(y_m)$ WITH $(y_m) = m \cdot 2^{-4}\sqrt{2/\pi}$

m	$f(y_m)$	m	$f(y_m)$	m	$f(y_m)$
1	0.960202	23	0.251362	45	0.024790
2	0.920502	24	0.231326	46	0.021754
3	0.881000	25	0.212486	47	0.019052
4	0.841870	26	0.194746	48	0.016642
5	0.803050	27	0.178118	49	0.014504
6	0.764710	28	0.162600	50	0.012614
7	0.727012	29	0.148096	51	0.010942
8	0.689890	30	0.134600	52	0.009470
9	0.653500	31	0.122076	53	0.008178
10	0.617986	32	0.110504	54	0.007044
11	0.583274	33	0.099796	55	0.006052
12	0.549500	34	0.089932	56	0.005190
13	0.516784	35	0.080884	57	0.004436
14	0.485044	36	0.072574	58	0.003784
15	0.454390	37	0.064976	59	0.003220
16	0.424918	38	0.058060	60	0.002732
17	0.396538	39	0.051754	61	0.002310
18	0.369332	40	0.046030	62	0.001950
19	0.343370	41	0.040860	63	0.001640
20	0.318554	42	0.036180	64	0.001376
21	0.294944	43	0.031964		
22	0.272580	44	0.028184		

By setting $\delta y = 2^{-r}\sqrt{2/\pi}$, we can find a suitable index $\hat{m}$ for $E_n^k(i,j)$ such that

$$\hat{m} \leqslant 2^r E_n^k(i,j)/\mu < \hat{m} + 1. \tag{14}$$

Then, we can use $f(y_{\hat{m}})$ to approximate $P(|Z| > E_n^k(i,j))$.

The effect of δy on the performance of the algorithm is investigated by varying δy from $2^{-1}\sqrt{2/\pi}$ to $2^{-8}\sqrt{2/\pi}$ (i.e., from $r = 1$ to $r = 8$) and measuring the PSNR performance of the algorithm. The result is shown in Fig. 5. It can be seen that the performance of the algorithm significantly improves as r increases but eventually saturates. The saturation of performance results from the fact that μ is also a random variable. If the variation of μ exceeds the step size, any further reduction of the step size can no longer provide more precise weights. Since the saturation starts roughly at $r = 4$, we choose $\delta y = 2^{-4}\sqrt{2/\pi}$ and construct a table for $f(y_m)$, as shown in Table I. This table is used to determine the interpolation weights in the experiments described in Section IV.

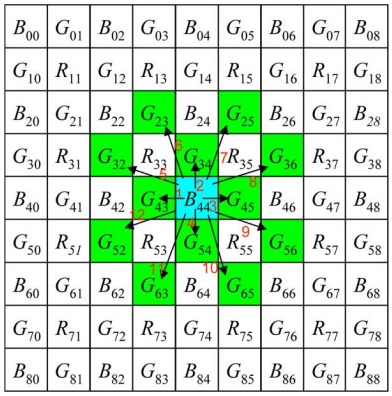

Fig. 6. Candidate pixels for interpolating the missing G at B_{44}.

TABLE II
DISPLACEMENTS FROM CURRENT PIXEL TO CANDIDATES IN STEP I

n	v^1_n	h^1_n	n	v^1_n	h^1_n
1	0	−1	7	−2	+1
2	−1	0	8	−1	+2
3	0	+1	9	+1	+2
4	+1	0	10	+2	+1
5	−1	−2	11	+2	−1
6	−2	−1	12	+1	−2

The stochastic weighting and missing color estimation procedure of is summarized as follows.

1) Calculate μ as defined in (13).
2) For each edge indicator $E^k_n(i,j)$, find index $\hat{m}$ such that condition (14) is satisfied.
3) Temporarily set $w^k_n(i,j) = f(y_{\hat{m}})$. Note that at this point the weights thus obtained do not satisfy condition (8) yet. The normalization is done in the next step.
4) Estimate the missing color $x^k(i,j)$ by

$$\hat{x}^k(i,j) = \sum_{n=1}^{N_k} \frac{x^k_n(i,j) w^k_n(i,j)}{\sum_{n=1}^{N_k} w^k_n(i,j)}. \tag{15}$$

C. Color Interpolation

Now we describe the details of the proposed algorithm. To exploit the high correlation between the R, G, and B channels, the color difference technique described in [4] is adopted in the algorithm.

1) Interpolate the Missing G of R and B Pixels: In this step, the missing G component of a pixel (e.g., B_{44} in Fig. 6) is interpolated from twelve nearby neighbors of the pixel. The vertical and horizontal positions of these twelve candidates relative to the pixel to be interpolated are specified in Table II. The edge indicator of each candidate pixel is obtained by setting $d = 2\sqrt{(v^1_1)^2 + (h^1_1)^2} = 2$ and substituting $k = 1, N_k = 12$, and

$$\kappa_n = \begin{cases} 1, & 1 \leqslant n \leqslant 4 \\ 0.5, & 5 \leqslant n \leqslant 12 \end{cases} \tag{16}$$

into (4). Here, κ_n is set to 0.5 for $n = 5, \ldots, 12$ to achieve the computational efficiency described in Section II. The color difference value of a blue pixel at (i,j) is estimated from its nth candidate by

$$b^1_n(i,j) = G\left(i + v^1_n, j + h^1_n\right) - \widehat{B}\left(i + v^1_n, j + h^1_n\right) \tag{17}$$

TABLE III
MOVES FROM CURRENT PIXEL TO CANDIDATES IN STEP II

n	v^2_n	h^2_n	n	v^2_n	h^2_n
1	−1	−1	3	+1	+1
2	−1	+1	4	+1	−1

TABLE IV
DISPLACEMENTS FROM CURRENT PIXEL TO CANDIDATES IN STEP III

n	v^3_n	h^3_n	n	v^3_n	v^3_n
1	0	−1	7	−2	+1
2	−1	0	8	−1	+2
3	0	+1	9	+1	+2
4	+1	0	10	+2	+1
5	−1	−2	11	+2	−1
6	−2	−1	12	+1	−2

where $\widehat{B}(i+v^1_n, j+h^1_n)$ is the average of the two blue pixels right next to $G(i+v^1_n, j+h^1_n)$. As an example, in Fig. 6, $\widehat{B}(2,3) = (B(2,2) + B(2,4))/2$ and $b^1_6(4,4) = G(2,3) - \widehat{B}(2,3)$.

The missing G value of the blue pixel is interpolated by

$$G(i,j) = B(i,j) + \frac{\sum_{n=1}^{12} w^1_n(i,j) b^1_n(i,j)}{\sum_{n=1}^{12} w^1_n(i,j)} \tag{18}$$

where the weights $w^1_n(i,j)$ are obtained by the method described in Section III-B.

Similarly, the missing G value of a red pixel located at (i,j) is interpolated by

$$G(i,j) = R(i,j) + \frac{\sum_{n=1}^{12} w^1_n(i,j) r^1_n(i,j)}{\sum_{n=1}^{12} w^1_n(i,j)} \tag{19}$$

where $r^1_n(i,j)$ denotes the color difference value of the red pixel estimated from the nth candidate pixel

$$r^1_n(i,j) = G\left(i + v^1_n, j + h^1_n\right) - \widehat{R}\left(i + v^1_n, j + h^1_n\right). \tag{20}$$

The G values thus obtained are used to calculate the color difference values in the following steps.

2) Interpolate the Missing $R(B)$ of $B(R)$ Pixels: In this step, the missing color of a pixel is interpolated from the four diagonal neighbors of the pixel. The vertical and horizontal positions of these four candidates relative to the pixel to be interpolated are specified in Table III.

Because the distance from the pixel to each candidate is the same, the edge indicators are calculated by substituting $\kappa_n = 1, k = 2$, and $N_k = 4$ into (4). Given $G(i+v^2_n, j+h^2_n)$ obtained in Step 1, the color difference values are computed by

$$b^2_n(i,j) = G\left(i + v^2_n, j + h^2_n\right) - B\left(i + v^2_n, j + h^2_n\right) \tag{21}$$
$$r^2_n(i,j) = G\left(i + v^2_n, j + h^2_n\right) - R\left(i + v^2_n, j + h^2_n\right) \tag{22}$$

and the weight for each candidate is computed using the method described in Section III-B. The missing B and R are interpolated by

$$B(i,j) = G(i,j) - \frac{\sum_{n=1}^{4} w^2_n(i,j) b^2_n(i,j)}{\sum_{n=1}^{4} w^2_n(i,j)} \tag{23}$$

$$R(i,j) = G(i,j) - \frac{\sum_{n=1}^{4} w^2_n(i,j) r^2_n(i,j)}{\sum_{n=1}^{4} w^2_n(i,j)}. \tag{24}$$

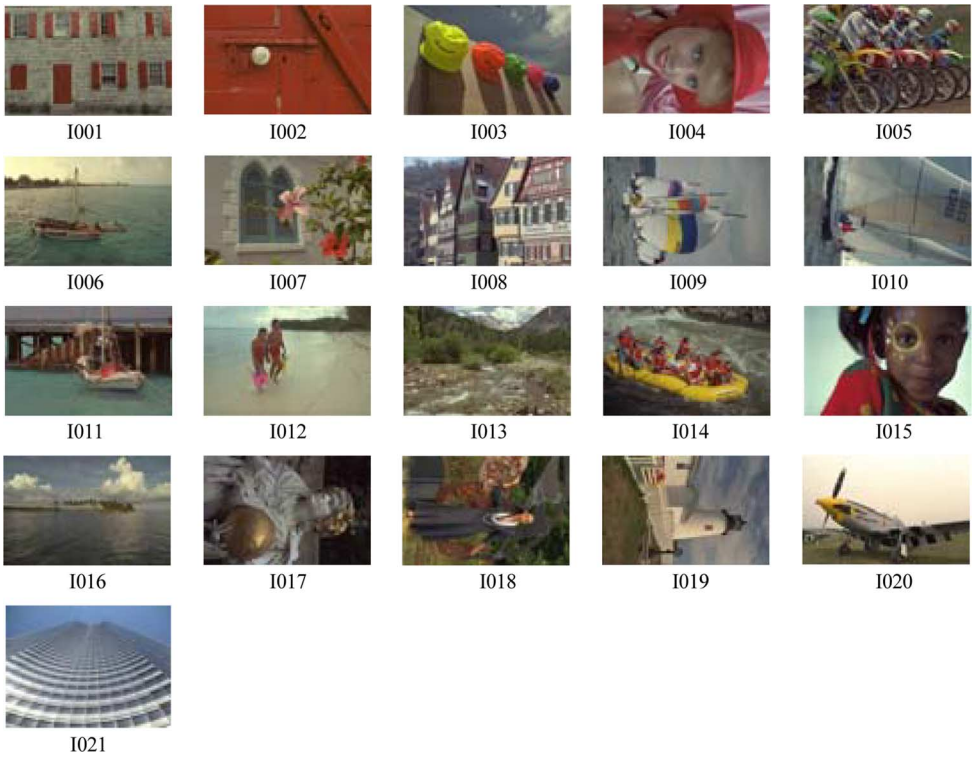

Fig. 7. Test images.

3) Interpolate the Missing R and B of G Pixels: In this step, each missing color of the pixel is interpolated from twelve candidates specified in Table IV. The edge indicators are obtained by substituting $k = 3, N_k = 12$, and κ_n defined in (16) into (4). The color difference values are obtained by

$$b_n^3(i,j) = G\left(i + v_n^3, j + h_n^3\right) - B\left(i + v_n^3, j + h_n^3\right) \quad (25)$$

$$r_n^3(i,j) = G\left(i + v_n^3, j + h_n^3\right) - R\left(i + v_n^3, j + h_n^3\right). \quad (26)$$

Then the missing colors are interpolated by

$$B(i,j) = G(i,j) - \frac{\sum_{n=1}^{12} w_n^3(i,j) b_n^3(i,j)}{\sum_{n=1}^{12} w_n^3(i,j)} \quad (27)$$

$$R(i,j) = G(i,j) - \frac{\sum_{n=1}^{12} w_n^3(i,j) r_n^3(i,j)}{\sum_{n=1}^{12} w_n^3(i,j)}. \quad (28)$$

IV. Experiments and Discussion

The test images used in our experiments consist of 20 images (labeled I001–I020) from the Kodak image database and a building image (I021) that has fine, dense edges in most areas of the image. These full-color test images, shown in Fig. 7, were first down-sampled to the Bayer pattern and then interpolated to full color. The performance of the various interpolation methods are measured in terms of peak signal-to-noise ratios (PSNRs) of the R, G, and B channels, mean square error (MSE), and normalized color difference (NCD). The first two metrics are performed in the RGB space, while the last one is in the CIELAB color space.

The PSNR of the G channel of an image with height H and width W is obtained by

$$\mathrm{PSNR}_G = 10 \cdot \log_{10}\left(\frac{255^2 \cdot H \cdot W}{\sum_{i=1}^{H}\sum_{j=1}^{W}(G(i,j) - G'(i,j))^2}\right) \quad (29)$$

where $G(i,j)$ and $G'(i,j)$ represent the green components of the original and the interpolated images, respectively. The PSNRs of the R and B channels are calculated in a similar way.

The MSE of the interpolated image is given by

$$\mathrm{MSE} = \frac{1}{3 \cdot H \cdot W}\sum_{i=1}^{H}\sum_{j=1}^{W}((\Delta R(i,j))^2 + (\Delta G(i,j))^2 + (\Delta B(i,j))^2) \quad (30)$$

where Δ denotes the difference between the original color value and the interpolated color value.

The NCD of the interpolated image is computed according to the formula shown in (31) at the bottom of the page. Because the square root operation is performed before the summation in

$$\mathrm{NCD} = \frac{\sum_{i=1}^{H}\sum_{j=1}^{W}\left(\sqrt{(\Delta L(i,j))^2 + (\Delta a(i,j))^2 + (\Delta b(i,j))^2}\right)}{\sum_{i=1}^{H}\sum_{j=1}^{W}\left(\sqrt{(L(i,j))^2 + (a(i,j))^2 + (b(i,j))^2}\right)}. \quad (31)$$

TABLE V
COLOR PSNR OF THE INTERPOLATED IMAGES

Image	GB			CDB			C2D2			AH			Proposed			Proposed with MF		
No.	PSNR_B	PSNR_G	PSNR_R	PSNR_B	PSNR_G	PSNR_R	PSNR_B	PSNR_G	PSNR_R	PSNR_B	PSNR_G	PSNR_R	PSNR_B	PSNR_G	PSNR_R	PSNR_B	PSNR_G	PSNR_R
I001	31.858	32.425	31.800	33.921	36.775	32.992	34.033	34.975	34.272	34.934	36.902	34.598	37.653	39.869	36.500	38.400	40.771	35.999
I002	37.384	38.253	36.273	40.025	42.410	25.878	39.562	40.825	37.311	40.132	42.504	37.668	41.317	43.377	35.601	40.776	41.920	35.094
I003	39.070	39.374	38.855	33.848	44.293	32.668	40.245	42.147	40.657	43.271	45.243	41.781	41.430	44.318	39.518	39.640	41.895	38.694
I004	37.819	38.096	37.145	35.100	42.622	33.300	40.663	41.086	38.823	40.623	42.124	38.798	41.337	42.189	36.758	40.878	39.950	35.814
I005	32.063	31.760	31.935	34.634	37.778	32.170	35.176	34.965	35.104	36.710	38.041	35.604	37.350	39.326	35.490	36.087	38.369	34.726
I006	33.094	33.865	33.243	35.241	38.144	33.568	34.496	36.062	34.816	37.941	39.816	37.651	37.952	40.425	36.953	38.628	41.858	37.621
I007	39.074	39.054	38.590	33.089	43.020	28.993	41.752	42.098	40.402	41.983	43.621	40.714	41.308	43.489	38.399	39.369	41.297	37.575
I008	30.371	31.153	30.318	31.105	33.921	30.835	31.759	33.266	31.731	33.383	35.777	32.984	33.587	36.442	32.784	34.124	37.629	33.135
I009	38.227	39.011	38.492	36.955	42.431	38.944	40.056	41.495	40.680	41.151	43.768	41.678	40.852	44.125	41.682	39.823	43.094	40.850
I010	37.517	38.290	37.882	38.277	43.086	37.926	39.843	40.965	40.272	41.203	43.675	41.248	41.121	44.191	41.111	40.467	43.127	40.213
I011	34.156	34.555	33.898	35.778	39.078	34.236	36.321	37.144	35.812	37.763	39.494	36.726	39.351	41.527	36.403	39.417	41.512	36.190
I012	39.230	39.891	38.894	39.154	43.571	36.812	40.727	42.069	40.246	42.835	44.978	41.740	42.360	44.753	40.284	41.546	43.570	39.196
I013	27.548	28.021	27.920	30.687	33.613	32.297	30.264	30.773	30.941	31.185	32.911	31.443	35.044	36.813	35.134	35.657	38.754	36.016
I014	34.065	34.230	33.009	32.472	38.999	32.399	36.228	37.041	34.952	37.446	38.996	34.706	37.292	39.218	33.134	35.724	37.754	32.249
I015	36.774	37.136	35.624	38.750	42.537	31.396	39.209	40.058	37.359	39.611	41.398	37.180	40.688	43.237	36.525	39.887	40.856	35.389
I016	36.700	37.546	36.773	36.478	41.042	37.720	37.589	39.412	37.923	41.674	43.559	41.585	39.928	43.090	39.660	40.382	43.747	40.217
I017	36.011	36.463	36.763	38.647	41.505	38.955	38.633	39.399	39.605	39.142	41.175	39.909	40.454	43.392	40.088	40.109	42.591	40.350
I018	31.630	32.374	32.432	33.162	37.627	30.451	34.047	35.067	35.263	34.445	36.678	35.172	36.512	39.505	36.606	36.054	39.171	36.302
I019	35.093	35.918	35.407	33.006	38.265	35.402	36.515	38.136	36.950	38.174	40.363	38.393	37.376	40.168	39.940	37.601	40.435	37.329
I020	36.294	36.893	36.798	36.848	41.439	36.821	38.518	39.500	38.938	39.038	41.327	39.300	40.751	42.807	38.267	39.440	41.977	38.406
I021	25.728	26.340	25.478	27.708	30.213	26.169	28.129	28.889	27.725	29.498	31.134	28.861	30.195	32.082	29.079	30.425	33.095	29.290
Avg.	34.747	35.270	34.644	34.995	39.637	33.330	36.846	37.875	36.656	38.197	40.166	37.511	38.755	41.160	36.996	38.306	40.627	36.698

Original Data: {10, 10, 10, 10, 2, 2, 2, 2}
Estimated Data 1: {11, 12, 11, 11, 1, 1, 1, 1}
Estimated Data 2: {11, 11, 13, 11, 1, 2, 2, 3}

	MSE	NCD
Data 1	1.375	0.1875
Data 2	1.75	0.1667

Fig. 8. Example of the inconsistency between MSE and NCD.

this equation, the NCD metric may not necessarily be consistent with MSE and PSNR. An example of the inconsistency is illustrated in Fig. 8, where the MSE of the first estimated data set is smaller than that of the second one, but the NCD gives the opposite result.

The conversion from RGB to CIELAB space may also contribute to the inconsistency, since it is not an orthogonal operation. Generally, if interpolation method A leads to a smaller MSE but larger NCD than method B for an image, the interpolation error of the image generated by A tends to have smaller variance than that by B, implying that A is more stable. Because the proposed algorithm is based on the locally stationary assumption, it is more region-independent than the other methods and hence is able to achieve minimal MSE.

The stochastic interpolation algorithm proposed in this paper is compared against four existing methods: gradient-based interpolation (GB), color-difference-based interpolation (CDB) [4], CFA interpolation using correlations and directional Derivatives (C2D2) [5], and adaptive homogeneity-directed interpolation (AH) [8]. The GB, CDB, and C2D2 methods are described in Section I. To investigate the effect of median filtering on the performance of the proposed algorithm, a 3×3 median filter (MF) [14] is applied as a post processing operation, and the result is compared.

The AH method proposed by Hirakawa *et al.* [8] applies a five point 1-D filter to the raw sensory data to generate a vertically interpolated image and a horizontally interpolated image. Then the homogeneity of each pixel of the two images is calculated. At each (i, j), the pixel from the image with the larger homogeneity is chosen as the output of the interpolation. Since it has been demonstrated that the AH method has better performance than the Alternating Projection method [9], we only include the former in the comparison. Note also that we chose to compare against the original AH method instead of the simplified versions [8] because it has the best performance.

The PSNR values of the images interpolated by different methods are listed in Table V, whereas the MSE and NCD values are provided in Table VI. As we can see, the AH method and the proposed algorithm have the best performance among all methods. Furthermore, the average performance of the proposed algorithm is better than that of AH by about 1 dB in PSNR_G and 13% in MSE. However, AH has a slightly better NCD. This probably can be attributed to the fact that AH uses a criterion similar to NCD in calculating the homogeneity.

Another interesting observation based on the average performance is that the median filter does not bring improvement to the interpolation. By examining the interpolated images carefully, however, we find that images with regions of dense edges

TABLE VI
MSE AND NCD OF INTERPOLATED IMAGES

Image	GB		CDB		C2D2		AH		Proposed		Proposed with MF	
No.	MSE	NCD(%)	MSE	NCD(%)	MSE	NCD(%)	MSE	NCD(%)	MSE	NCD(%)	MSE	NCD(%)
I001	40.89	3.329	24.23	2.692	23.57	2.558	18.90	2.280	10.79	1.856	10.39	1.769
I002	12.31	2.018	59.40	4.158	8.22	1.775	7.03	1.618	8.56	2.011	9.91	2.143
I003	8.01	1.252	21.47	1.857	5.23	1.105	3.11	0.887	4.78	1.217	6.68	1.359
I004	11.12	1.741	18.02	2.052	6.39	1.411	6.06	1.328	7.47	1.603	9.65	1.792
I005	41.81	3.931	24.23	3.221	20.18	2.888	13.99	2.446	12.64	2.515	15.79	2.685
I006	29.81	2.048	19.34	1.856	20.22	1.771	9.47	1.214	9.81	1.366	8.13	1.274
I007	8.38	1.203	39.05	2.904	4.76	1.034	4.15	0.962	5.71	1.302	7.91	1.467
I008	56.70	3.101	43.47	3.018	39.23	2.742	26.58	2.197	25.83	2.343	22.66	2.243
I009	9.05	1.189	8.37	1.089	5.53	0.962	4.05	0.860	4.09	0.830	5.10	0.874
I010	10.58	1.309	7.78	1.284	6.02	1.046	4.20	0.936	4.18	0.943	5.06	1.023
I011	24.75	2.574	16.58	2.348	14.93	2.184	10.67	1.754	9.00	1.923	9.22	1.939
I012	7.61	0.899	8.10	0.924	5.23	0.780	3.27	0.637	4.01	0.729	5.08	0.762
I013	107.31	5.333	40.71	3.399	55.99	3.832	43.14	3.470	17.95	2.327	14.20	2.093
I014	27.52	2.783	27.48	2.502	16.38	2.186	13.97	1.958	17.17	2.110	22.53	2.277
I015	14.68	1.709	19.81	1.939	8.72	1.410	8.09	1.315	7.70	1.417	10.27	1.584
I016	13.00	1.738	10.25	1.666	9.76	1.559	3.94	1.076	5.61	1.335	4.96	1.311
I017	14.89	2.190	7.25	1.649	7.83	1.700	6.51	1.556	5.07	1.618	5.31	1.692
I018	39.82	4.164	33.75	4.135	21.74	3.072	19.03	2.980	12.00	2.621	13.08	2.776
I019	18.50	1.928	20.33	1.996	12.54	1.628	8.43	1.374	10.44	1.530	9.73	1.518
I020	14.05	0.981	10.54	0.948	8.25	0.832	6.85	0.775	6.19	0.891	6.97	0.919
I021	169.71	4.088	109.75	4.158	97.94	3.321	69.20	2.634	60.94	2.831	55.81	2.715
Avg.	32.40	2.358	27.14	9.724	18.98	1.895	13.84	1.631	11.90	1.682	12.31	1.725

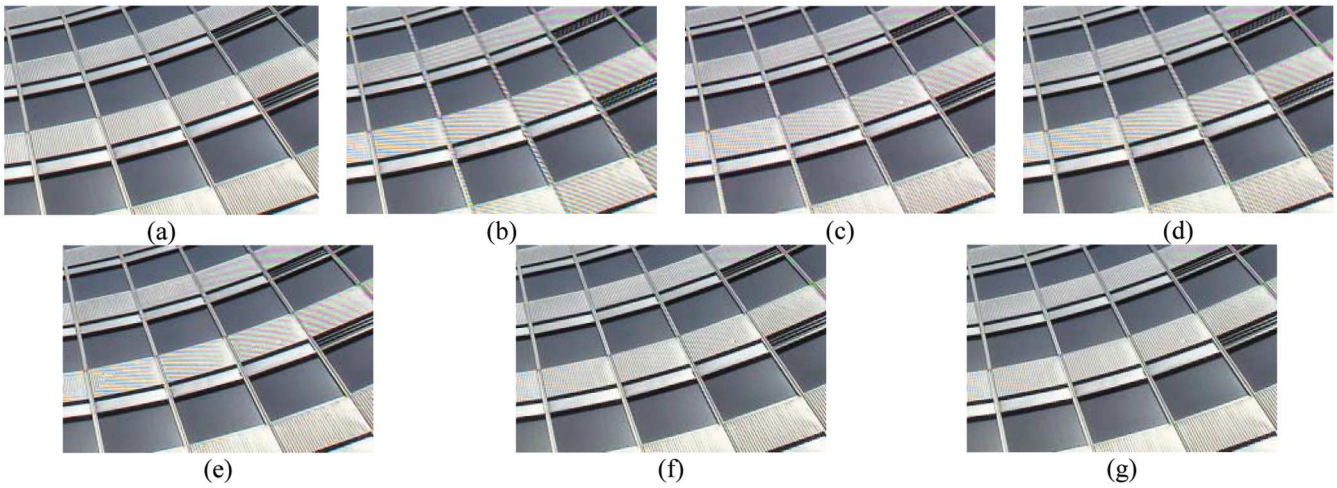

Fig. 9. Part of the building image (I021). (a) Original. (b) GB. (c) CDB. (d) C2D2. (e) AH. (f) Proposed. (g) Proposed with MF.

such as the building image (I021), the light house image (I019), and the brook image (I013) do benefit from the media filtering. For images with large smooth areas, the median filter tends to degrade the overall performance of the interpolation, though it is able to remove the aliasing near the edges.

The robustness of the proposed algorithm against aliasing is demonstrated in Figs. 9–12. It is evident from Fig. 9 that the proposed algorithm is able to deal with dense edges and reduce the perceptible aliasing effectively. Similar fine results can be observed at the sticker near the center of Fig. 10, the wires and trees in Fig. 11, and the slogan and mascot in Fig. 12. The results show that the algorithm produces better quality for the interpolated images by preserving the sharpness of object boundaries and by effectively preventing the occurrences of false colors.

The proposed algorithm provides significant performance improvement without additional computational costs. Although it needs to compute the average of the edge indicators and to perform table look-up, it does not need to compute the reciprocal of the edge-indicator as the C2D2 algorithm does. The overall computational cost of the proposed algorithm is similar to that of other soft-decision edge-adaptive methods.

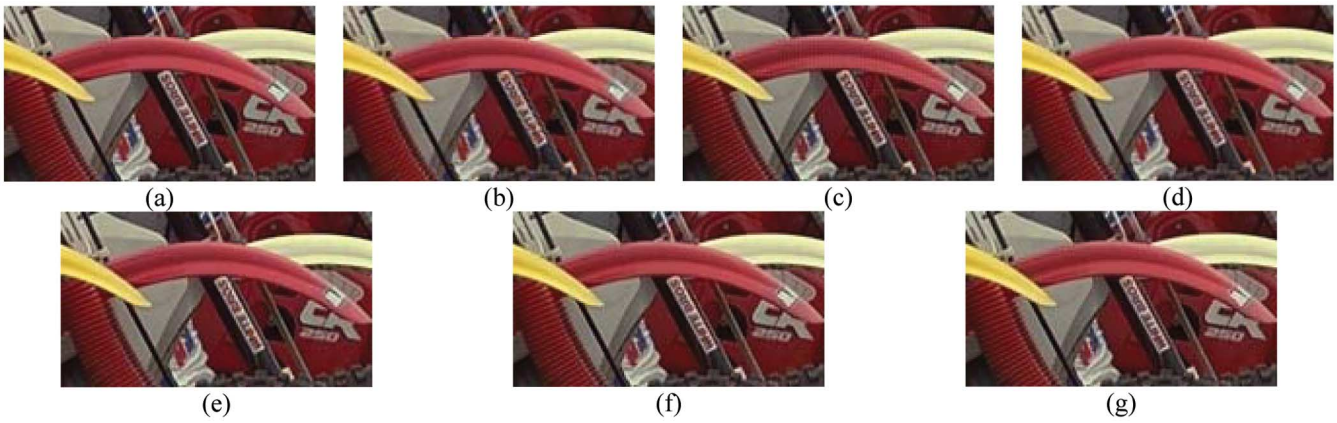

Fig. 10. Part of the motors image (I005). (a) Original.(b) GB.(c) CDB.(d) C2D2.(e) AH.(f) Proposed.(g) Proposed with MF.

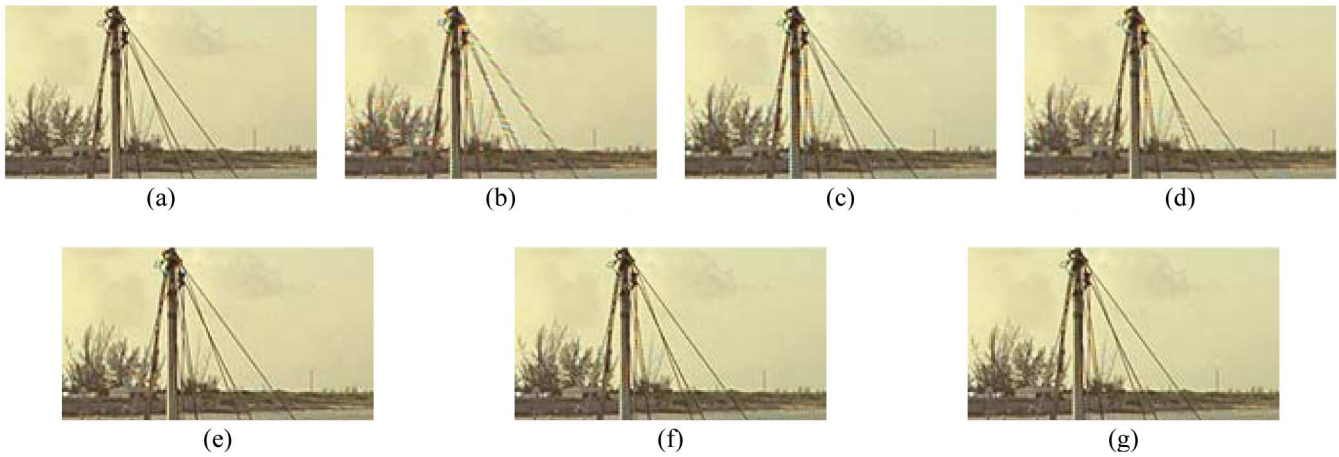

Fig. 11. Part of the sailboat image (I006). (a) Original.(b) GB.(c) CDB.(d) C2D2.(e) AH.(f) Proposed.(g) Proposed with MF.

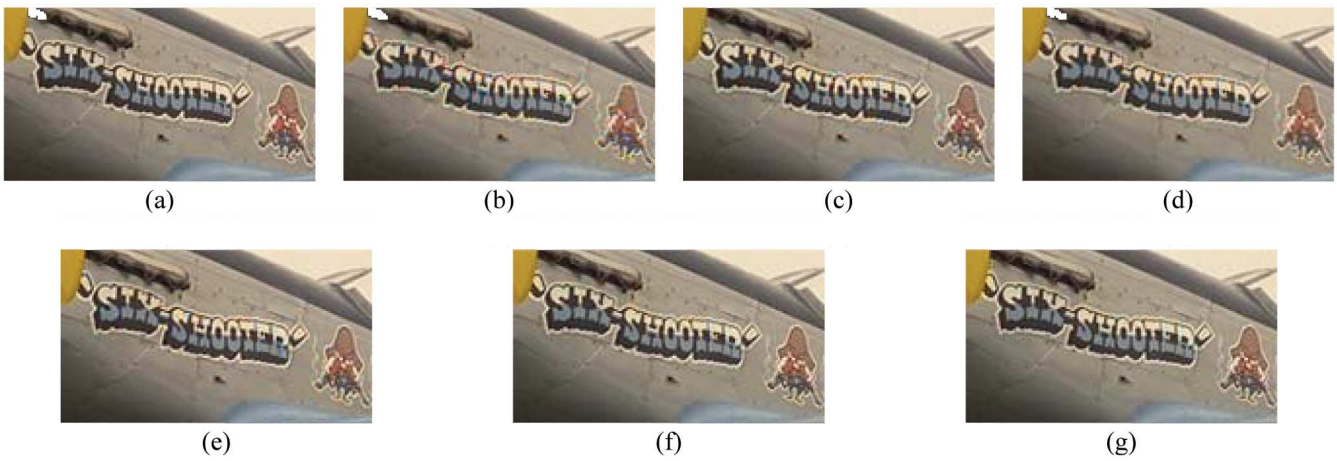

Fig. 12. Part of the airplane image (I020). (a) Original. (b) GB. (c) CDB. (d) C2D2. (e) AH. (f) Proposed. (g) Proposed with MF.

V. Summary

In this paper, we have presented a stochastic estimation algorithm for adaptive color interpolation of the Bayer pattern. Each missing color of a pixel is interpolated from a set of uniformly oriented candidate pixels. By using the properties of the image under the locally stationary Gaussian process assumption and a stochastic weighting policy, the algorithm demonstrates high edge sensing ability and robust weighting against aliasing.

References

[1] B. E. Bayer, "Imaging array," U.S. Patent 3 971 065, Jul. 20, 1976.
[2] D. R. Cok, "Signal processing method and apparatus for producing interpolated chrominance values in sampled color image signal," U.S. Patent 4 642 678, 1987.
[3] X. Li and M. T. Orchard, "New edge-directed interpolation," *IEEE Trans. Image Process.*, vol. 10, no. 10, pp. 1521–1527, Oct. 2001.
[4] S. C. Pei and I. K. Tam, "Effective color interpolation in CCD color filter arrays using signal correlation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 6, pp. 503–513, Jun. 2003.

[5] N. Kehtarnavaz, H. Oh, and Y. Yoo, "Color filter array interpolation using correlations and directional derivatives," *J. Electron. Imag.*, vol. 12, no. 4, pp. 621–632, Oct. 2003.
[6] R. Ramanath, W. E. Synder, and G. L. Bilbro, "Demosiacking methods for Bayer color array," *J. Electron. Imag.*, vol. 11, no. 3, pp. 306–315, Jul. 2002.
[7] B. S. Hur and M. G. Kang, "High definition color interpolation scheme for progressive scan CCD image sensor," *IEEE Trans. Consum. Electron.*, vol. 47, no. 1, pp. 179–186, Feb. 2001.
[8] K. Hirakawa and T. W. Parks, "Adaptive homogeneity-directed demosaicing algorithm," *IEEE Trans. Image Process.*, vol. 14, no. 3, Mar. 2005.
[9] 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.
[10] R. Ramanath and W. E. Snyder, "Adaptive demosaicking," *J. Electron. Imag.*, vol. 12, no. 4, pp. 633–642, Oct. 2003.
[11] W. Lu and Y. P. Tan, "Color filter array demosaicking: New method and performance measure," *IEEE Trans. Image Process.*, vol. 12, no. 10, pp. 1194–1210, Oct. 2003.
[12] R. Lukac, K. N. Plataniotis, D. Hatzinakos, and M. Aleksic, "A novel cost effective demosaicing approach," *IEEE Trans. Consum. Electron.*, vol. 50, no. 1, pp. 256–261, Feb. 2004.
[13] R. Lukac and K. N. Plataniotis, "Data adaptive filters for demosaicking: A framework," *IEEE Trans. Consum. Electron.*, vol. 51, no. 2, pp. 560–570, May 2005.
[14] T. W. Freeman, "Median Filter for Reconstructing Missing Color Samples," U.S. Patent 4 724 395, 1988.
[15] B. K. Gunturk, J. Glotzbach, Y. Altunbasak, R. W. Schafer, and R. M. Mersereau, "Demosaicking: Color filter array interpolation," *IEEE Signal Process. Mag.*, vol. 22, no. 1, pp. 44–54, Jan. 2005.
[16] R. Lukac, K. Martin, and K. N. Plataniotis, "Demosaicked image processing using local color ratios," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 14, no. 6, pp. 914–920, Jun. 2004.
[17] R. Lukac and K. N. Plataniotis, "Normalized color ratio modeling for CFA interpolation," *IEEE Trans. Consum. Electron.*, vol. 50, no. 2, pp. 737–745, May 2004.
[18] X. Wu and N. Zhang, "Primary-consistent soft-decision color demosaicking for digital cameras," *IEEE Trans. Image Process.*, vol. 13, no. 9, pp. 1263–1274, Sep. 2004.
[19] D. Alleysson, S. Stisstrunk, and J. Hérault, "Linear demosaicing inspired by the human visual system," *IEEE Trans. Image Process.*, vol. 14, no. 4, pp. 439–449, Apr. 2005.
[20] C. A. Laroche and M. A. Prescott, "Apparatus and method for adaptively interpolating a full color image utilizing chrominance gradients," U.S. Patent 5 373 322, Dec. 12, 1994.
[21] J. F. Hamilton, Jr. and J. E. Adams Jr., "Adaptive color plane interpolation in single sensor color electronic camera," U.S. Patent 5 629 734, May 13, 1997.
[22] E. Chang, S. Cheung, and D. Pan, "Color filter array recovery using a threshold-based variable number of gradients," *Proc. SPIE*, vol. 3650, pp. 36–43, Mar. 1999.
[23] R. Kimmel, "Demosaicing: Image reconstruction from color CCD samples," *IEEE Trans. Image Process.*, vol. 8, no. 9, pp. 1221–1228, Sep. 1999.
[24] H. S. Malvar, L.-W. He, and R. Culter, "High-quality linear interpolation for demosaicing of Bayer-patterned color images," in *Proc. ICASSP*, May 2004, pp. 485–488.

Hung-An Chang received the B.S. degree in electrical engineering from National Taiwan University, Taiwain, R.O.C., in 2004. He is currently pursuing the Ph.D. degree in the Computer Science and Artificial Intelligence Laboratory, Massachusetts Institute of Technology, Cambridge, and working on the application of digital signal processing and machine learning techniques to speech processing and spoken language understanding.

Homer H. Chen (S'83–M'86–SM'01–F'03) received the Ph.D. degree in electrical and computer engineering from University of Illinois at Urbana-Champaign.

Since August 2003, he has been with the College of Electrical Engineering and Computer Science, National Taiwan University, Taiwan, R.O.C., where he is Irving T. Ho Chair Professor. Prior to that, he had held various research and development management and engineering positions in leading US companies including AT&T Bell Labs, Rockwell Science Center, iVast, and Digital Island over a period of 17 years. He was a US delegate of the ISO and ITU standards committees and contributed to the development of many new interactive multimedia technologies that are now part of the MPEG-4 and JPEG-2000 standards. His research interests lie in the broad area of multimedia processing and communications.

Dr. Chen is an Associate Editor of IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY. Hehas served as an Associate Editor for IEEE TRANSACTIONS ON IMAGE PROCESSING from 1992 to 1994, Guest Editor for IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY in 1999, and Editorial Board Member for *Pattern Recognition* from 1989 to 1999.