Design of practical color filter array interpolation algorithms for digital cameras

James E. Adams, Jr.

Eastman Kodak Company, Imaging Research and Advanced Development
Rochester, New York 14653-5408

ABSTRACT

Single-chip digital cameras use a color filter array and subsequent interpolation strategy to produce full–color images. While the design of the interpolation algorithm can be grounded in traditional sampling theory, the fact that the sampled data is distributed among three different color planes adds a level of complexity. Previous ways of treating this problem were based on computationally intensive approaches, such as iteration. Such methods, while effective, cannot be implemented in today's crop of digital cameras due to the limited computing resources of the cameras and the accompanying host computers. These previous methods are usually derived from general numerical methods that do not make many assumptions about the nature of the data. Significant computational economies, without serious losses in image quality, can be achieved if it is recognized that the data is *image* data and some appropriate image model is assumed. To this end, the design of practical, high–quality color filter array interpolation algorithms based on a simple image model is discussed.

Keywords: color filter array interpolation, digital cameras, image model

1. INTRODUCTION

In a previous paper,[1] the author described the make-up of a digital camera image processing chain with particular emphasis on the color filter array (CFA) interpolation process. This paper again explores the CFA interpolation process, this time from the standpoint of Fourier spectrum analysis and optimum algorithm design.

Though the work presented can be generalized to most any CFA pattern, for simplicity, the Kodak Bayer CFA pattern[2] will be assumed throughout this paper. Figure 1 is an illustration of this pattern. In Fig. 1, R stands for red, G stands for green, and B stands for blue.

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

Figure 1. Bayer CFA Pattern

This paper will concentrate on the reconstruction of the luminance information in the image. In this case, the green pixel information will be treated as the luminance information. The subsequent reconstruction of the red and blue information will be performed using Cok's method, described in the author's previous paper.[1,3]

A one-dimensional approach to CFA interpolation will be used. Used in conjunction with an adaptive strategy for selecting either a horizontal or vertical one-dimensional pixel neighborhood for each pixel in question, produces a CFA interpolation algorithm capable of very high-quality image reconstructions. It will be assumed that the appropriate orientation for interpolation has already been chosen. The task at hand will be to show how to use the pixels within the resulting one-dimensional slice to produce the best estimate for the missing green pixel value in question.

Further author information –
Email: jeadams@kodak.com; Telephone: 716-253-6506; Fax: 716-726-3927

In order to design an optimum green pixel value predictor, the interpolation problem will be stated as a simple signal sampling and recovery problem, after Gaskill.[4] This will establish the characteristics of the "perfect" CFA interpolation predictor. Subsequently, practical approximations to this ideal predictor will be developed and the performances of these approximate predictors compared to the ideal performance to understand the compromises incurred.

It should be realized that the image processing operation modeled here, CFA sampling and interpolation, is not shift-invariant. Therefore, the following analysis, which assumes a linear, shift-invariant (LSI) system, cannot be expected to be rigorously correct. However, it does provide a framework from which some general and pragmatic results can be derived. The reader is reminded to keep this caveat in mind.

2. SAMPLING THEORY REVIEW

We will assume the one-dimensional pixel neighborhood in Fig. 2 throughout the rest of this paper.

$$\cdots \quad R_{-2} \quad G_{-1} \quad R_0 \quad G_1 \quad R_2 \quad \cdots$$

Figure 2. One-dimensional Pixel Neighborhood

In Fig. 2, R would be either a red pixel (for red-green rows) or a blue pixel (for blue-green rows) and G would be a green pixel. If f(x) is the original green image information, then the sampled green data, $f_S(x)$, would be given in Eq. 1.

$$f_S(x) = \sum_{n=-\infty}^{\infty} f(1+2n)\delta(x-1-2n) \tag{1}$$

The Fourier transform of Eq. 1 is given in Fig. 2.

$$F_S(\xi) = \frac{1}{2}\sum_{n=-\infty}^{\infty} (-1)^n F\left(\xi - \frac{n}{2}\right) \tag{2}$$

Equation 2 indicates the well-known fact that the spectrum of the sampled signal consists of the spectrum of the original signal, F(ξ), replicated along the frequency axis at regular intervals. Assuming the spectrum replicates do not overlap, then by eliminating the spectrum replicates, F(ξ), and, therefore, f(x), can be recovered. The ideal interpolation filter to perform such a spectrum replicate elimination is

$$\mathrm{Intp}(\xi) = \mathrm{rect}(2\xi), \tag{3}$$

where

$$\mathrm{rect}\left(\frac{x-x_0}{b}\right) = \begin{cases} 0, & \left|\frac{x-x_0}{b}\right| > \frac{1}{2} \\ \frac{1}{2}, & \left|\frac{x-x_0}{b}\right| = \frac{1}{2}. \\ 1, & \left|\frac{x-x_0}{b}\right| < \frac{1}{2} \end{cases} \tag{4}$$

Applying Intp(ξ) to $F_S(\xi)$ produces the desired result, to within an unimportant multiplicative constant. (See Eq. 5.)

$$F_I(\xi) = F_S(\xi)\mathrm{Intp}(\xi) = \frac{1}{2}F(\xi) \tag{5}$$

Taking the inverse Fourier transform of Eq. 5 produces the ideal interpolation process. (See Eq. 6.)

$$f_I(x) = \frac{1}{2}\sum_{n=-\infty}^{\infty} f(1+2n)\,\mathrm{sinc}\left(\frac{x-1-2n}{2}\right) = \frac{1}{2}f(x) \tag{6}$$

In Eq. 6, Gaskill's definition of the sinc function is assumed:

$$\mathrm{sinc}\left(\frac{x-x_0}{b}\right) = \frac{\sin\left[\pi\left(\frac{x-x_0}{b}\right)\right]}{\pi\left(\frac{x-x_0}{b}\right)}. \tag{7}$$

Of course, Eq. 6 is the well-known Shannon-Whittaker sampling theorem.[5] Equally well known is that a direct implementation of Eq. 6 is impractical because the summation has a painfully slow rate of convergence.

Evaluating Eq. 6 for odd integral values (i.e., green pixels) and for even integral values (i.e., red or blue pixels) produces the desired results:

$$\begin{aligned} f_I(m) &= \frac{1}{2}f(m), && \text{m odd} \\ &= \frac{1}{2}\left\{\begin{array}{l} [f(m-1)+f(m+1)]\,\mathrm{sinc}\left(\frac{1}{2}\right) \\ \quad +[f(m-3)+f(m+3)]\,\mathrm{sinc}\left(\frac{3}{2}\right)+\cdots \end{array}\right\}, && \text{m even} \end{aligned} \tag{8}$$

3. ANALYSIS OF AVERAGING NEAREST NEIGHBORS

Since Eq. 6 does not converge quickly enough for practical use, an approximation must be employed. The simplest approximation for interpolating Fig. 2 is to average the nearest green neighbors. This would be equivalent to convolving the sampled green data with the finite impulse response (FIR) filter given in Eq. 9.

$$h = \frac{1}{2}\begin{pmatrix}\frac{1}{2} & 1 & \frac{1}{2}\end{pmatrix} \tag{9}$$

(The one-half scaling factor is included to be consistent with Eq. 6.) The equivalent intp(x) function of Eq. 9 is

$$\mathrm{intp}(\mathrm{x}) = \frac{1}{2}\left[\delta(x) + \frac{1}{2}\delta\delta(x)\right], \tag{10}$$

where

$$\delta\delta\left(\frac{x-x_0}{b}\right) = |b|\left[\delta(x-x_0+b)+\delta(x-x_0-b)\right]. \tag{11}$$

The equivalent of Eq. 8 using Eq. 10 is

$$f_I(m) = \frac{1}{2} f(m), \qquad \text{m odd}$$
$$= \frac{1}{2}\left\{\frac{f(m-1)+f(m+1)}{2}\right\}, \qquad \text{m even} \tag{12}$$

The Fourier transform of Eq. 10 is

$$\text{Intp}(\xi) = \cos^2(\pi\xi). \tag{13}$$

Figure 3 is a plot of Eqs. 3 and 13. It can be seen that nearest neighbor average spectrum departs significantly from the ideal case.

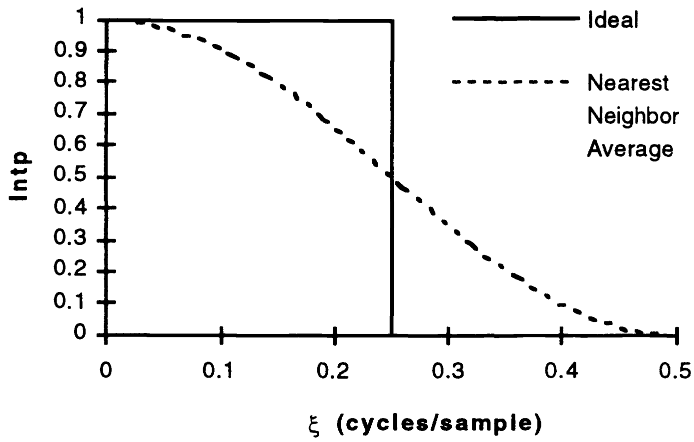

Figure 3. Comparison of Ideal and Nearest Neighbor Average Interpolating Function Spectra

Figure 4 is an example of the kind of distortion that can occur in an image using the nearest neighbor average of Eq. 10.

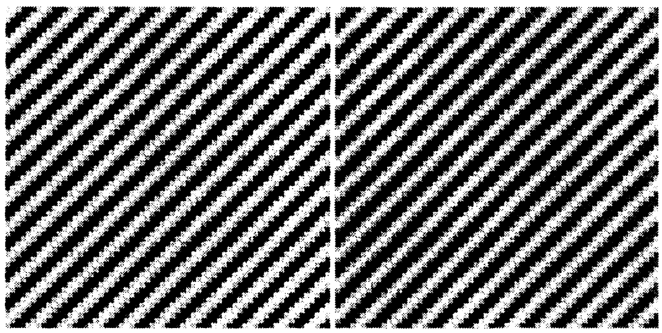

Figure 4. (Left) Original Image, (Right) Interpolated Image using Nearest Neighbor Averaging.

4. ENLARGING THE INTERPOLATION NEIGHBORHOOD

One way to improve upon Eq. 10 is to simply expand the pixel neighborhood and use more green pixel values. There are problems with this approach. First, the number of rows of image data that must be held in memory at one time (it is assumed you want to preserve the option of interpolating either horizontally or vertically for any given pixel location) becomes prohibitive, from a memory usage standpoint. Second, because green pixels occur only at every other pixel location, half of the possible frequencies that can be used to construct Intp(ξ), are not available. To illustrate this last point, consider the following infinite impulse response (IIR) filter:

$$h = \frac{1}{2}(\cdots \quad a_2 \quad 0 \quad a_1 \quad 1 \quad a_1 \quad 0 \quad a_2 \quad \cdots). \tag{14}$$

The equivalent interpolating function and its Fourier transform are given below.

$$\text{intp}(x) = \frac{1}{2}\left[\delta(x) + \sum_{n=1}^{\infty} \frac{a_n}{2n-1} \delta\delta\left(\frac{x}{2n-1}\right)\right] \tag{15}$$

$$\begin{aligned} \text{Intp}(\xi) &= \frac{1}{2}\left\{1 + 2\sum_{n=1}^{\infty} a_n \cos[2(2n-1)\pi\xi]\right\} \\ &= \frac{1}{2}[1 + 2a_1\cos(2\pi\xi) + 2a_2\cos(6\pi\xi) + \cdots] \end{aligned} \tag{16}$$

As can be seen from Eq. 16, $\cos(4\pi\xi)$, $\cos(8\pi\xi)$, etc., are missing.

5. INTRODUCTION OF AN IMAGE MODEL

One solution to the problems of the previous section would be to use the red and blue pixels in the prediction of missing green values. To do this, something about the correlation between color channels needs to be assumed. One well-known image model is to simply assume that red and blue are perfectly correlated to green over the extent of the interpolation pixel neighborhood. This image model will be stated as

$$G_n = R_n + k \tag{17}$$

where n refers to the pixel location, R would be the red pixel value, G would be the green pixel value, and k would be the appropriate bias for the given pixel neighborhood. As in Fig. 2, R could also stand for a blue pixel value.

As an illustration of the reasonability of Eq. 17, Fig. 5 shows a small image and its decomposition into green, red - green and blue - green planes. As can be seen, the contrast of the red - green and blue - green images is quite flat over most of the scene.

Figure 5. Upper Left: Original (rendered as a gray scale image), Upper Right: Green Channel, Lower Left: Red - Green, Lower Right: Blue - Green

6. DESIGNING AN IMPROVED FAMILY OF PREDICTORS

To take advantage of Eq. 17, the Eq. 9 FIR filter is generalized to a 5-point FIR filter:[6]

$$h = \frac{1}{2}\left[\begin{pmatrix} a_1 & 1 & a_1 \end{pmatrix} + \begin{pmatrix} a_2 & 0 & a_0 & 0 & a_2 \end{pmatrix}\right]. \tag{18}$$

In Eq. 18, the first 3-point kernel is convolved with the existing green pixel values. The second 5-point kernel is convolved with the existing red pixel values. The corresponding interpolating function is given in Eq. 19.

$$\mathrm{intp}(x) = \frac{1}{2}\left\{\delta(x) + a_0\left[\delta(x) + k\right] + a_1\delta\delta(x) + a_2\left[\frac{1}{2}\delta\delta\left(\frac{x}{2}\right) + 2k\right]\right\} \tag{19}$$

Note the introduction of k in Eq. 19 is a result of employing Eq. 17 to allow the mixing of the green pixel values and red pixel values.

The first issue to address is the factor k. Since k will rarely, if ever, be known, it is best to constrain the values of a_0 and a_2 so that k is eliminated from Eq. 19. This results in the following constraint:

$$a_0 + 2a_2 = 0. \tag{20}$$

Another constraint will be to normalize Intp(ξ) at x equals zero, i.e.,

$$\mathrm{Intp}(0) = 1. \tag{21}$$

To determine appropriate values for a_0, a_1, and a_2, the first step is to take the Fourier transform of Eq. 19:

$$\mathrm{Intp}(\xi) = \frac{1}{2}\left\{1 + a_0(1+k) + 2a_1\cos(2\pi\xi) + a_2\left[2\cos(4\pi\xi) + 2k\right]\right\}. \tag{22}$$

Using Eqs. 20 and 21, Eq. 22 may be reduced to

$$\mathrm{Intp}(\xi) = \cos^2(\pi\xi) + a_0\sin^2(2\pi\xi) \tag{23}$$

Note that when a_0 is set to zero, then Eqs. 23 and 18 (with the help of Eq. 20) reduce to Eqs. 13 and 9, respectively. Figure 6 is a family of curves created by using Eq. 23 and a range of values for a_0. The ideal response of Eq. 3 is included as well.

Concentrating on the frequency range below 0.25 cycles/sample, it appears the preferred value for a_0 is in the vicinity of 0.3. Three candidate values for a_0 will be selected for subsequent image processing experimentation: zero, one-third and one-half. These produce the three FIR interpolation kernels given in Eqs. 24, 25, and 26.

$$h = \frac{1}{2}\begin{pmatrix} \frac{1}{2} & 1 & \frac{1}{2} \end{pmatrix} \tag{24}$$

$$h = \frac{1}{2}\left[\begin{pmatrix} \frac{1}{2} & 1 & \frac{1}{2} \end{pmatrix} + \begin{pmatrix} -\frac{1}{6} & 0 & \frac{1}{3} & 0 & -\frac{1}{6} \end{pmatrix}\right] \tag{25}$$

$$h = \frac{1}{2}\left[\begin{pmatrix} \frac{1}{2} & 1 & \frac{1}{2} \end{pmatrix} + \begin{pmatrix} -\frac{1}{4} & 0 & \frac{1}{2} & 0 & -\frac{1}{4} \end{pmatrix}\right] \tag{26}$$

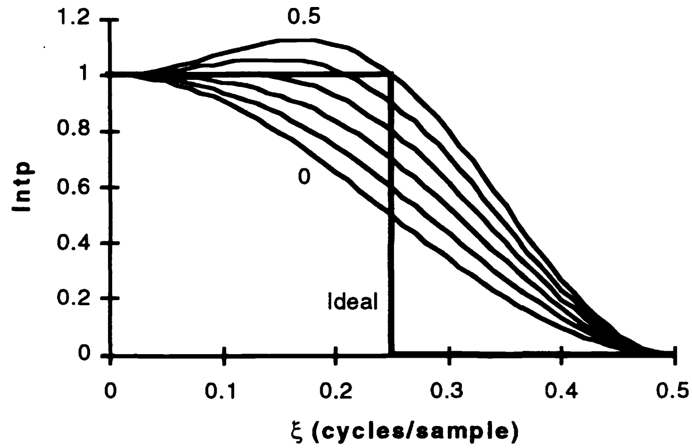

Figure 6. Family of Curves Based on Eq. 23 for Values of a_0 From 0 to 0.5 Every 0.1

7. EXAMPLE IMPLEMENTATIONS

Figure 7 shows the results of taking a diagonal sinusoidal pattern (one-sixth cycles/sample along both the horizontal and vertical frequency axes) and interpolating it with Eqs. 24 through 26.

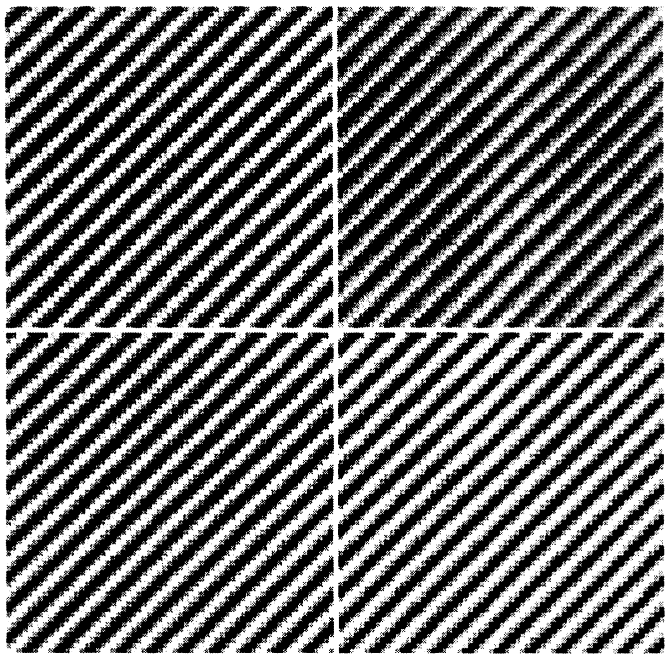

Figure 7. Upper Left: Original (1/6 cycles/sample), Upper Right: Eq. 24 Interpolation, Lower Left: Eq. 25 Interpolation, Lower Right: Eq. 26 Interpolation

The results shown in Fig. 7 seem to correspond with the Intp(ξ) curves in Fig. 6. The upper left quadrant is the original image. The upper right quadrant was processed using Eq. 24 (a_0 equals zero). According to Fig. 6, a distorted reproduction with lower contrast than the original might be expected. This is what is seen in Fig. 7. The lower left quadrant was processed with Eq. 25 (a_0 equals one-third). According to Fig, 6, a nearly perfect reconstruction is called for. The difference between the upper left quadrant and lower left quadrant is hard to see, if it is visually detectable at all. The lower right quadrant was processed with Eq. 26 (a_0 equals one-half). According to Fig. 6, a distorted image with higher contrast than the original is suggested. The lower right quadrant does, indeed, appear to have a higher contrast than the original. It appears for this case that there is a good match between Figs. 6 and 7.

Incidentally, the upper quadrants of Fig. 7 are the same as the ones shown in Fig. 4 except that a small phase shift has been added to the original prior to processing. While the overall impression of the results of applying Eq. 24 is the same, the detailed construction is quite different between Figs. 4 and 7. This presents a reminder that the interpolation process as discussed in this paper is not a shift-invariant system. Therefore, we should not expect the analysis presented in this work to be an exact representation of what actual image processing demonstrates.

Figure 8 shows the results of taking a section from a typical image (of a lighthouse) and interpolating it with Eqs. 24 through 26.

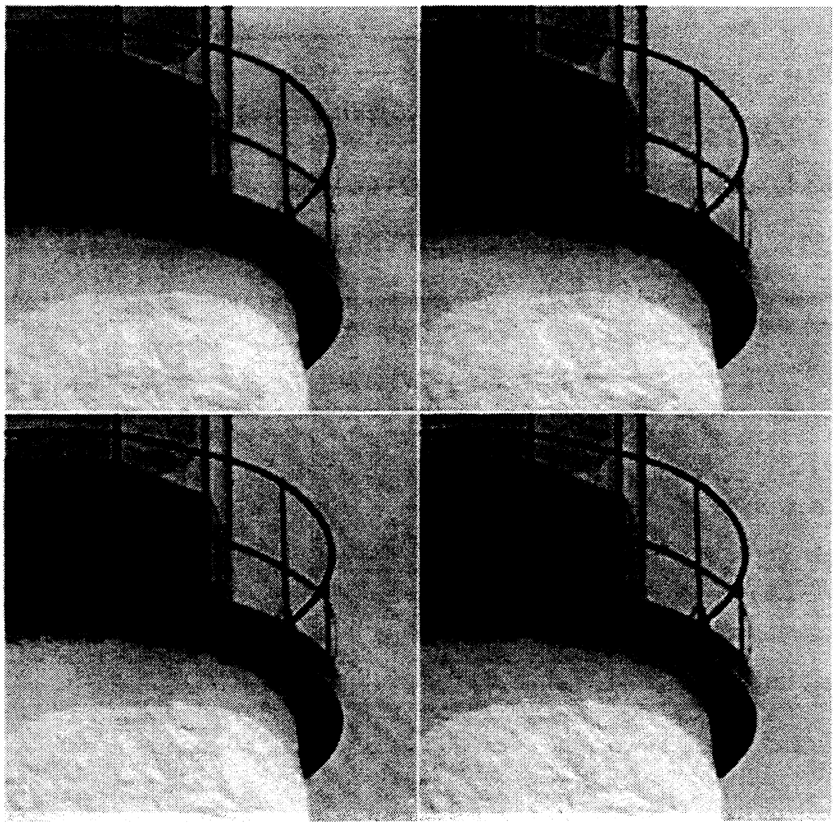

Figure 8. Upper Left: Original, Upper Right: Eq. 24 Interpolation, Lower Left: Eq. 25 Interpolation, Lower Right: Eq. 26 Interpolation. All images rendered as gray scales.

Unlike in Fig. 7, the images in Fig. 8 have gone through an entire digital camera image processing chain (e.g., color and tone correction, sharpening, etc.) The results of Fig. 8 still mirror the results of Fig. 7 very closely. There are a number of obvious artifacts and other distortions along the lighthouse railing in the upper right quadrant (Eq. 24 processing). There is very little visual difference between the upper left quadrant and the lower left quadrant (Eq. 25 processing). The lower right quadrant (Eq. 26 processing) appears to have a slightly higher contrast than the upper left quadrant and there are some minor artifacts along the lighthouse railing, as well. When these images are viewed in color, the differences within Fig. 8 described here are more pronounced. Once again, there appears to be a good match between Fig. 6 and Fig. 8.

8. SUMMARY

The process of color filter array sampling and interpolation can be cast in the form of a signal sampling and recovery problem using standard Fourier spectrum analysis. A simple image model can be used to permit the use of information from color channels other than green to aid in the reconstruction of the green (luminance) record. Fourier spectra can be derived for various CFA interpolation kernels and relative image quality predictions be made from these spectra. Actual image processing simulations tend to support that validity of this approach.

ACKNOWLEDGMENTS

The author would like to thank John Hamilton, Kevin Spaulding, and Brian Keelan, all of Eastman Kodak Company, for their valuable contributions to this material.

REFERENCES

1. J. E. Adams, Jr., "Interactions between color plane interpolation and other image processing functions in electronic photography", *Proceeding of SPIE*, C. Anagnostopoulos, M. Lesser, eds., vol. 2416, pp. 144–151, SPIE, Bellingham, WA, 1995.
2. B. E. Bayer, "Color imaging array", U.S. Patent 3,971,065, 1976.
3. D. R. Cok, "Signal processing method and apparatus for producing interpolated chrominance values in a sampled color image signal", U.S. Patent 4,642,678, 1987.
4. J. D. Gaskill, *Linear systems, Fourier Transforms, and Optics*, John Wiley & Sons, New York, p. 266, 1978.
5. *Ibid.*, p. 271.
6. J. E. Adams, Jr., J. F. Hamilton, Jr., "Adaptive color plane interpolation in single sensor color electronic camera", U.S. Patent 5,506,619, 1996.