Interactions between color plane interpolation and other image processing functions in electronic photography

James E. Adams, Jr.

Eastman Kodak Company, Imaging Research and Advanced Development
Rochester, New York 14650-1816

ABSTRACT

Electronic cameras using a single CCD detector acquire scene color by subsampling in three, color planes and subsequently interpolating the information to reconstruct three, full-resolution color planes. The nature and size of the interpolation errors are a function of the algorithm used. When interpolation errors are propagated through the rest of the imaging chain, it becomes evident that synergistic effects among image processing operations must be considered when selecting and tuning an interpolation algorithm. This presentation demonstrates and comments on these image processing interactions.

1. INTRODUCTION

1.1. Spectral image capture

In a photographic imaging system three planes of color information are captured simultaneously. Each cyan, magenta, and yellow layer contains the full spatial content for the associated spectral band. Creating a strictly analogous CCD imaging system presents a number of challenges. To simulate photography's simultaneous capture of three planes of color information would require three separate imaging detectors. The current cost and packaging considerations that accompany a three-detector system make this approach impractical for all but the most extreme situations. The cost per CCD is high and the necessity of dividing the incident light among multiple detectors reduces the overall signal-to-noise of the system in a significant way.

1.2. Bayer color filter array

The solution is to use a single CCD and to cover its surface with a filter mosaic called a color filter array, or CFA. Each filter in the CFA covers a single pixel and passes only a specific spectral band.

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

Figure 1. Bayer CFA Pattern.

G1	R2	G3

Figure 2. Example Pixel Neighborhood

The CFA illustrated in Fig. 1 is called the Bayer pattern. Each pixel is covered with a filter that transmits either red, green, or blue light. Working with this particular CFA will be the topic of this paper.

1.3. Interpolation overview

The problem with a single sensor Bayer CFA system is that each pixel produces information about only one of the three spectral bands. For each pixel two colors are "missing". Before any further image processing can occur, estimates of the missing color information for each pixel must be made. These estimates are usually made by interpolating neighboring pixel values. In Fig. 2, a red pixel is flanked on either side by green pixels. One could estimate the missing green value, G2, by taking the average of the flanking pixels as given in Equation 1.

$$G2 = \frac{G1 + G3}{2} \qquad (1)$$

The interpolation problem is more complex than this example may suggest. Localized spatial features such as edges can cause estimates made with forms of Equation 1 to be quite inaccurate. This can lead to significant pixel artifacts in the final image.

0-8194-1763-7/95/$6.00

The general approach to improving interpolation estimates is to consider larger pixel neighborhoods and more complicated estimators. Additionally, one can employ adaptive algorithms that select a preferred estimator based on the spatial features present in the neighborhood. Finally, one cannot be content with performing the interpolation in isolation of the rest of the imaging chain. Certain types of residual errors of the interpolation process may be amplified significantly by subsequent image processing steps. Therefore, it becomes important to view any interpolation algorithm within the context of the entire imaging chain.

2. IMAGE PROCESSING CHAIN

2.1. Image capture

This paper will consider an electronic still camera system. This type of system has significant differences with video camera systems that will have fundamental impacts on the interpolation process. These effects will be discussed in detail in Sections 3 and 4. Figure 3 is a schematic of the image capture portion of the image chain that will be assumed.

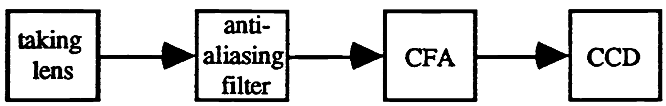

Figure 3. Schematic of Image Capture Portion of Imaging Chain

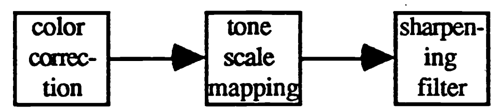

Figure 6. Schematic of Subsequent Imaging Processing Chain

The taking lens forms an image of the scene on the CCD. An anti-aliasing filter is inserted into the chain to prevent aliasing in the image due to the discrete spatial sampling nature of the CCD. The CFA will be assumed to be a Bayer pattern as illustrated in Fig. 1.

2.2. Interpolation

The image coming from the CCD will have the obvious effects of the CFA pattern. Figure 4 is a portion of an image captured with a system like the one in Fig. 3. The goal of the interpolation step will be to take the data represented in Fig. 4 and produce a full, three-color image like in Fig. 5.

2.3. Subsequent processing

Once the image has been interpolated, further processing is necessary to produce a final image. Figure 6 is a schematic of the remaining image processing chain. The image in Fig. 5 is typically desaturated due to being in the color space of the CCD and image capture optics (lens, anti-aliasing filter, and CFA). The first step after interpolation is to correct the color. The next step is to map the tone scale of the image for proper rendering on the final display device (i.e., CRT or printer). The final step is to sharpened the image to compensate for the effects of the anti-aliasing filter and other sources of blur. Figure 7 might be a typical result.

It is clear from comparing Figs. 5 and 7 that it is important to evaluate the results of the interpolation routine by looking at the final image of the entire image processing chain. Slight differences in adjacent pixels in Fig. 5 are highlighted and magnified in Fig. 7. For this reason, all the image processing algorithms explored below will be evaluated from the viewpoint of the final system image.

3. LINEAR EXPOSURE SPACE INTERPOLATION

3.1. Reference image and pixel neighborhood

Figure 8a is a digitized image that has been processed with our electronic imaging chain with the exception of the CFA and interpolation steps. (Since this image was scanned from a photographic original, all three color values for each pixel are known.) This will be our reference image. Note that this will be a difficult image to interpolate well because the vertical columns of the skyscraper are at a small angle to the CCD pixel grid. Figure 8b is a typical pixel neighborhood taken from near the center of the skyscraper. It is important to remember that Fig. 8b is only a typical neighborhood and not entirely representative of the range of possible pixel neighborhoods. It is best to evaluate any interpolation routine by considering entire images and many images.

3.2. Nearest neighbor replication

A common interpolation technique used in applications where speed is of the essence is replication of nearest neighbors.[1] Referring to Fig. 9, we can define the replication schemes in Equations 2 and 3.

B1	G2	B3
G4	R5	G6
B7	G8	B9

Figure 9. Reference Pixel Neighborhood

$$G5 = G4 \tag{2}$$
$$B5 = B1 \tag{3}$$

The remaining replication schemes can be defined using the same rule: move either one pixel to the left, one pixel up, or both one pixel to the left and one pixel up to get the required color value. The pixel code values are in terms of linear exposure space. Figure 10 is the result of using this approach. It can be seen that significant color errors occur throughout the neighborhood. In a video imaging application many of these problems would be "averaged away" in the several times a second refresh process. In a still imaging system the errors would be unacceptable.

3.3. Bilinear interpolation

The next approach is the use of bilinear interpolation.[2] Again, if the procedures are very simple then this scheme can be implemented to run quickly. Referring to Fig. 9, the following estimators can be defined.

$$G5 = \frac{G2 + G4 + G6 + G8}{4} \tag{4}$$
$$B5 = \frac{B1 + B3 + B7 + B9}{4} \tag{5}$$
$$B2 = \frac{B1 + B3}{2} \tag{6}$$
$$B4 = \frac{B1 + B7}{2} \tag{7}$$

Variable substitution in Equations 4 through 7 produce all the remaining estimators. Equations 4 and 5 could be simplified to be only two point averages. For example, one could use just G4 and G6 to predict G5. Figure 11 is the result of this processing. The overall color and artifact errors are less extreme, but new pixel artifacts (the so-called "zipper effect") have been introduced into the neighborhood. Again, in a video application much of this may not be visible due to motion blur between video frames. In a still imaging application this would still not be acceptable.

3.4. Smooth hue transition interpolation

The key objection of the pixel artifacts in Figs. 10 and 11 is that the hues of adjacent pixels change abruptly and in an unnatural manner. The Bayer CFA pattern can be thought of as consisting of a luminance channel (the relatively numerous green pixels) and a chrominance channel (the relatively sparse red and blue pixels). A scheme can be created that interpolates these channels differently. The luminance channel can be interpolated as before (Equation 4). The chrominance channel can be interpolated by imposing a smooth transition in hue from pixel to pixel.[3] To do this, a blue "hue value" is defined as in Equation 8. A red "hue value" would be analogously defined.

$$\frac{B}{G} \tag{8}$$

Referring to Fig. 9, new blue estimators can be defined.

$$B2 = \frac{G2}{2}\left(\frac{B1}{G1} + \frac{B3}{G3}\right) \tag{9}$$

$$B4 = \frac{G4}{2}\left(\frac{B1}{G1} + \frac{B7}{G7}\right) \tag{10}$$

$$B5 = \frac{G5}{4}\left(\frac{B1}{G1} + \frac{B3}{G3} + \frac{B7}{G7} + \frac{B9}{G9}\right) \tag{11}$$

It is assumed that the luminance channel interpolation has already been performed so that all the green pixel values are defined. Similar estimators would be used for estimating red pixel values. Figure 12 was created using this approach. Again, an improvement has been made. The colors of the pixel artifacts have been desaturated significantly. In general, desaturating a pixel artifact will make it less objectionable in the final image.

4. LOGARITHMIC EXPOSURE SPACE INTERPOLATION

4.1. Smooth hue transition interpolation

Until now we have worked with pixel data in linear exposure space. There are advantages in interpolation work if the pixel data is transformed into logarithmic exposure space first.[3] Instead of Equation 8, one can now define the "hue value" as in Equation 12.

$$B - G \tag{12}$$

Note that B and G are now logarithmic exposure values. Following the idea of Equations 9 through 11, one can define Equations 13 through 15. Corresponding predictors for red are similar.

$$B2 = G2 + \frac{B1 - G1 + B3 - G3}{2} \tag{13}$$

$$B4 = G4 + \frac{B1 - G1 + B7 - G7}{2} \tag{14}$$

$$B5 = G5 + \frac{B1 - G1 + B3 - G3 + B7 - G7 + B9 - G9}{4} \tag{15}$$

Note that Equations 13 through 15 are not exact mathematical transforms of Equations 9 through 11. Figure 13 is the result of using Equations 13 through 15. Note that luminance interpolation is still being done with Equation 4 in linear exposure space. Figure 12 and Fig. 13 are nearly identical. The biggest difference was in the creation of these images. Equations 13 through 15 are far more computationally efficient and can be easily implemented using integer arithmetic and binary shifts. As a result, it takes less time to generate a Fig. 13 than a Fig. 12.

4.2. Adaptive algorithms

The main type of artifact in Fig. 13 is "zipper". This occurs because the luminance interpolation is averaging neighboring pixels indiscriminately. Referring to Fig. 9, there are a number of different ways to predict the missing pixel value G5 using some or all of the pixels {G2, G4, G6, G8}. If the interpolation algorithm can detect local spatial features present in the pixel neighborhood, then it can make effective choices as to which predictor to use for that neighborhood. The result is a reduction or elimination of "zipper-type" artifacts.

4.2.1. Edge sensing interpolation

One can create three different predictors for the missing green value G5.[4] These are given in Equations 16 through 18. (Equation 18 is identical to Equation 4.)

$$G5H = \frac{G4 + G6}{2} \quad (16)$$

$$G5V = \frac{G2 + G8}{2} \quad (17)$$

$$G5A = \frac{G2 + G4 + G6 + G8}{4} \quad (18)$$

To decide which of these predictors is appropriate for a given pixel neighborhood, we calculate the following two classifiers:

$$\Delta H = |G4 - G6| \quad (19)$$

$$\Delta V = |G2 - G8| \quad (20)$$

Finally, we define a threshold value, T. Equations 19 and 20 are the simplest ways of sensing if there is some kind of edge in the pixel neighborhood. The adaptive luminance interpolation algorithm can now be expressed in Fig. 14. Note that luminance interpolation still occurs in linear exposure space.

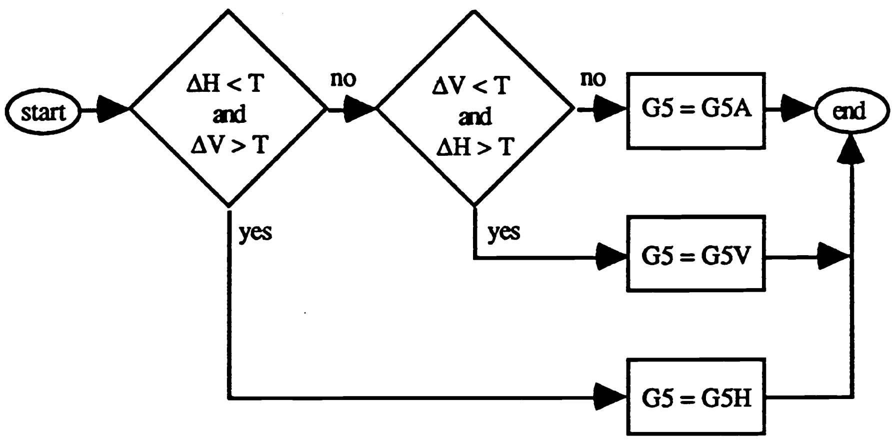

Figure 14. Adaptive Luminance Interpolation Algorithm

Figure 15 shows the result of combining the adaptive luminance interpolation of Fig. 14 with the logarithmic smooth hue chrominance interpolation of Equations 13 through 15. Figure 15 exhibits far less "zipper" in the neighborhood. This approach is sensitive to the chosen threshold value. Figure 16 was created with the same processing used to create Fig. 15 except that the value of the threshold was optimized for the neighborhood. It can be seen that the number of pixel artifacts has been reduced even more. Figure 16 represents a limit of sorts: reducing the threshold even more begins to increase the number of pixel artifacts again. It should be noted that different neighborhoods will have different optimum threshold values.

4.2.2. Pattern recognition interpolation

The edge sensing algorithm of the previous section can be generalized into a general pattern recognition approach to luminance interpolation. David Cok of Eastman Kodak Company has described such an approach.[5] Referring to Fig. 9, the pixel values of {G2, G4, G6, G8} can form four patterns (plus their rotations). These are diagrammed in Fig. 17.

(a)	(b)	(c)	(d)
H	*L*	*H*	*H*
H *H*	*H* *H*	*H* *L*	*L* *L*
L	*L*	*L*	*L*

Figure 17. The Four Luminance Pixel Patterns

H refers to a pixel with a code value greater than the average of all four pixels in the neighborhood. L refers to a pixel with a code value less than the average of all four pixels in the neighborhood. For convenience, Figs. 17a and 17d are called edge patterns, Fig. 17b is called a stripe pattern and Fig. 17c is called a corner pattern. Each pattern has its own predictor for the missing central green pixel. For the edge pattern, Equation 21 is used. The median is calculated by rank ordering the luminance pixel values so that A > B > C > D, where A is the largest pixel value among {G2, G4, G6, G8}, B is the next largest, etc., and then calculating the average of pixel values B and C.

$$G5 = M \equiv median\{G2, G4, G6, G8\} \tag{21}$$

For the stripe pattern a larger pixel neighborhood is needed to resolve the inherent ambiguity. (Figure 17b could be a stripe of low-value pixels in a neighborhood of high-value pixels or visa versa.) Figure 18 describes this larger neighborhood.

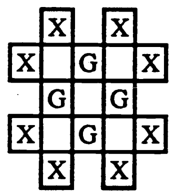

Figure 18. Stripe Pattern Neighborhood

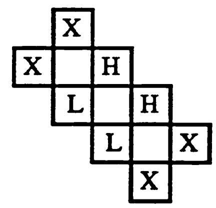

Figure 19. Corner Pattern Neighborhood

The predictor for this neighborhood is given in Equation 22.

$$G5 = clip_C^B(2M - S) \tag{22}$$

In Equation 22, M is the median of the pixels marked "G" and S is the average of the pixels marked "X". The clip function constrains the prediction to lie between the values B and C, where B and C are the rank order pixel values from the calculation of M. If the prediction is greater than B, it is set to B. If the prediction is less than C, it is set to C.

The corner pattern also uses an enlarged neighborhood. This neighborhood is described in Fig. 19. The predictor for this neighborhood is given in Equation 23.

$$G5 = clip_C^B(2M - S') \tag{23}$$

In Equation 23, M is the median of the pixels marked "H" and "L", and S' is the average of the pixels marked "X". The clip function is identical to the one used in Equation 22. Figure 20 is the result of using this pattern recognition approach to luminance interpolation. The pixel artifacts in the neighborhood have been significantly reduced without the need of defining a neighborhood sensitive threshold. There are still some remaining pixel artifacts in parts of the neighborhood, but when compared to the previous interpolation algorithm results, Fig. 20 seems to be the best overall.

5. CONCLUSIONS

A number of CFA interpolation routines have been investigated in the context of a given electronic still camera system. Typical video imaging algorithms such as pixel replication and simple bilinear interpolation appear to be inadequate for still imaging applications due to systematic hue errors and significant pixel artifacts. Smooth hue transition algorithms help to lessen systematic hue errors. Adaptive edge sensing interpolation greatly reduces pixel artifacts. A pattern recognition approach to luminance interpolation seems to perform the best of all the algorithms considered for the image at hand.

6. ACKNOWLEDGMENTS

The author would like to thank David Cok, Douglas Couwenhoven, John Hamilton, Robert Hibbard, and Kevin Spaulding, all of Eastman Kodak, for their fruitful discussions of this topic.

7. REFERENCES

1. U.S. Patent No. 4,716,455, N. Ozawa, et al, Hitachi, Ltd.
2. U.S. Patent No. 4,605,956, D. Cok, Eastman Kodak Company.

3. U.S. Patent No. 4,642,678, D. Cok, Eastman Kodak Company.
4. U.S. Patent Application No. 85,520, R. Hibbard, Eastman Kodak Company.
5. U.S. Patent No. 4,630,307, D. Cok, Eastman Kodak Company.

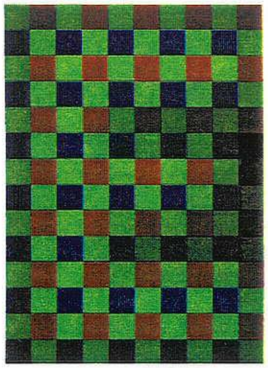

Figure 4. Portion of a Raw CCD Image

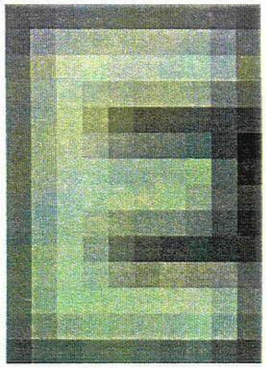

Figure 5. Portion of Interpolated Image

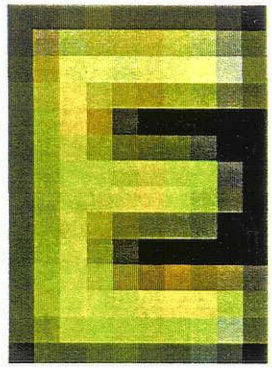

Figure 7. Portion of Fully Processed Image

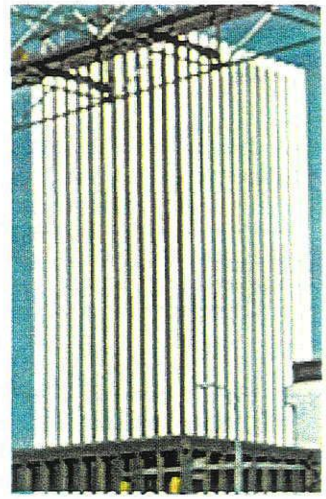

Figure 8a. Reference Image

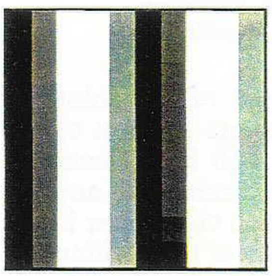

Figure 8b. Reference Image

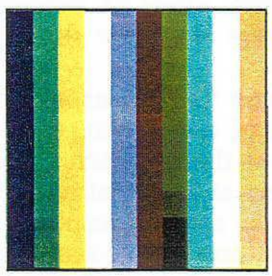

Figure 10. Pixel Replication Image

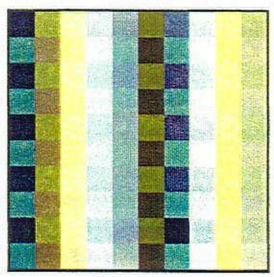

Figure 11. Bilinear Interpolation Image

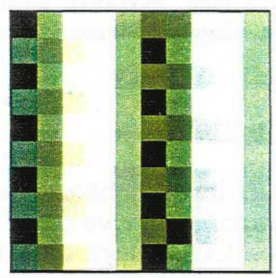

Figure 12. Linear Smooth Hue Image

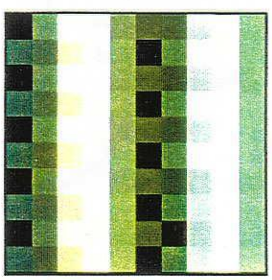

Figure 13. Logarithmic Smooth Hue Image

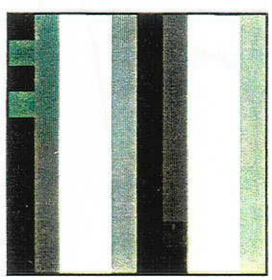

Figure 15. Adaptive Luminance Image, High Threshold

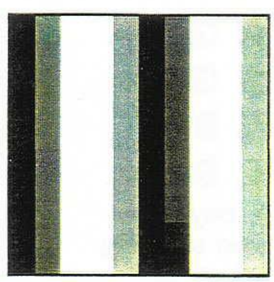

Figure 16. Adaptive Luminance Image, Low Threshold

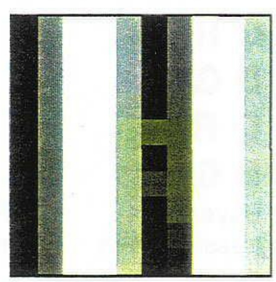

Figure 20. Pattern Recognition Image