RECONSTRUCTION OF COLOR IMAGES FROM CCD ARRAYS

D. Darian Muresan, Steve Luke and Thomas W. Parks

School of Electrical and Computer Engineering
Cornell University, Ithaca NY. 14853
darian, parks@ee.cornell.edu and scl9@cornell.edu

ABSTRACT

Digital color images from single chip digital cameras are obtained by interpolating a color filter array. In our paper we present an algorithm for interpolating the Bayer CFA (Color Filter Array) pattern. Color information is encoded by means of a CFA, which contains different color filters (i.e. red green and blue), placed in some pattern. The resulting sparsely sampled images of the three-color planes are interpolated to obtain dense images of the three-color planes and thus the complete color image. Interpolation usually introduces color artifacts due to the phase shifted, aliased signals introduced by the sparse sampling of the CFAs. We discuss a non-linear interpolation scheme based on edge information, that produces better visual results than those obtained by linear interpolation.

1. INTRODUCTION

Due to hardware limitations, CCD arrays in digital cameras do not capture the full red, green and blue color planes. Instead, they capture a sparsely sampled image of each of the color planes and interpolation is then used to reconstruct the original colors. In this paper we analyze the reconstruction of a single digital color image from the information provided by the Bayer CFA of Fig. 1. Previous work on this type of CFA interpolation can been found in [2, 4, 7] . In our work, we bring together some of these ideas, together with our own, to present a non-linear interpolation approach.

2. REVIEW

The Bayer array (Fig. 1) contains more green (or luminance) pixels than red or blue in order to provide high spatial frequency in luminance at the expense of chrominance signals. Our task is to interpolate each of the R, G and B planes.

The most basic idea is to independently interpolate the R, G and B planes. In other words, to find the missing green values use only neighboring green values, to find the missing blue values use only neighboring blue pixels and so on for red. More specifically, for linear interpolation, to obtain the missing green pixels, calculate the average of the four known neighboring green pixels. To calculate the missing blue pixels, proceed in two steps. First, calculate the missing blue pixels at the red location by averaging the four neighboring blue pixels. Second, calculate the missing blue pixels at the green locations by averaging the four neighboring blue pixels. The second step is equivalent to taking 3/8 of each of the closest pixels and 1/16 of four next closest pixels. This type of interpolation, which we call linear interpolation, introduces serious aliasing artifacts, as it can be seen in Fig. 4

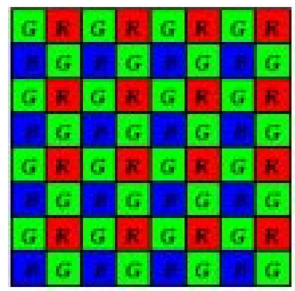

Fig. 1. Bayer Color Filter Array (CFA)

To obtain better interpolation, we would like to take out the effects of varying light intensity. With non-uniform lighting, since the green signal is essentially the same as luminance [1], the ratios of blue to green and red to green remain approximately constant within an object. This observation has been used by Cok [2] to improve interpolation and can be used here to improve our simple linear interpolation. Since green pixels are the most abundant, we first do a linear interpolation over the green pixels. Next, to obtain the missing blue pixels, we use the green pixels together with the known blue pixels to do a linear interpolation of the ratio of blue to green. Similarly, we use the ratio of red to green for interpolating the red channel (Fig. 5). This method improves the results of linear interpolation, but still

This work was supported by Texas Instruments

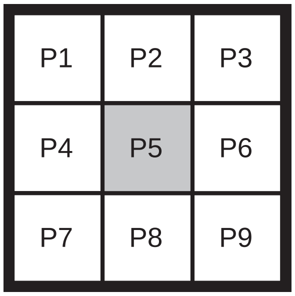

Fig. 2. Pixel P_5 is the pixel that we are trying to interpolate.

smears edges, since the green interpolation is a low pass filtering operation.

An additional improvement over this method is to interpolate along edges and not across them. Looking at Fig. 2, let E_i denote the likelihood that pixel P_5 and pixel P_i belong to the same edge. In other words, if P_5 and P_i are part of the same object then E_i is close to unity, otherwise it's close to zero. With this E_i, the improved interpolation algorithm would then be:

- First, interpolate the greens using only the green information:

$$G_5 = \frac{E_2 G_2 + E_4 G_4 + E_6 G_6 + E_8 G_8}{E_2 + E_4 + E_6 + E_8} \qquad (1)$$

- For the blues, we proceed in two steps. First interpolate the missing blues at the red locations

$$B_5 = G_5 \frac{E_1 \frac{B_1}{G_1} + E_3 \frac{B_3}{G_3} + E_7 \frac{B_7}{G_7} + E_9 \frac{B_9}{G_9}}{E_1 + E_3 + E_7 + E_9} \qquad (2)$$

  and second, interpolate the missing blues at the green locations.

$$B_5 = G_5 \frac{E_2 \frac{B_2}{G_2} + E_4 \frac{B_4}{G_4} + E_6 \frac{B_6}{G_6} + E_8 \frac{B_8}{G_8}}{E_2 + E_4 + E_6 + E_8} \qquad (3)$$

- Interpolate red pixels similarly to the blue pixels.

All that is left to do now, is to choose a proper E_i function. The interpolation proposed by Cok in [2] can be interpreted as choosing the function E_i as follows:
1). For green (i.e. we are trying to interpolate the green at pixel P_5):

$$\begin{cases} E_2 = E_8 = 1; E_4 = E_6 = 0, \text{ if } & \begin{array}{lcl} |G_4 - G_6| & < & T \\ |G_2 - G_8| & > & T \end{array} \\ E_2 = E_8 = 0; E_4 = E_6 = 1, \text{ if } & \begin{array}{lcl} |G_4 - G_6| & > & T \\ |G_2 - G_8| & < & T \end{array} \\ E_2 = E_4 = E_6 = E_8 = 1, \text{ else} & \end{cases}$$

where T is some predefined threshold value.

2). For red and blue (i.e. we are trying to interpolate the red or blue at location P_5):

$$E_1 = E_3 = E_7 = E_9 = 1;$$

$$\begin{cases} E_2 = E_8 = 1; E_4 = E_6 = 0 \text{ if } P_2 \text{ and } P_8 \text{ are blue in CFA} \\ E_2 = E_8 = 0; E_4 = E_6 = 1 \text{ if } P_4 \text{ and } P_6 \text{ are blue in CFA} \end{cases}$$

The results of this method can be seen in Fig. 6.

In [7], Kimmel developed a new function E by using gradient estimates based on differences. From Fig. 2 we define our gradients at pixel P_5, in the $x, y, x - diagonal$ and $y - diagonal$ directions as follows:

$$D_x(P_5) = \frac{P_4 - P_6}{2} \qquad D_y(P_5) = \frac{P_2 - P_8}{2}$$
$$D_{xd}(P_5) = \frac{P_3 - P_7}{2\sqrt{2}} \qquad D_{yd}(P_5) = \frac{P_1 - P_9}{2\sqrt{2}}$$

Notice that differences are always from the same color plane. If pixel P_5 is a green pixel then so is P_1, P_3, P_7 and P_9. To obtain the gradient in the diagonal directions at a green pixel, we can do a little better by defining our diagonal gradients as:

$$\begin{aligned} D_{xd}(P_5) &= \max\left\{\left|\frac{P_3 - P_5}{\sqrt{2}}\right|, \left|\frac{P_7 - P_5}{\sqrt{2}}\right|\right\} \\ D_{yd}(P_5) &= \max\left\{\left|\frac{P_1 - P_5}{\sqrt{2}}\right|, \left|\frac{P_9 - P_5}{\sqrt{2}}\right|\right\} \end{aligned}$$

With the above gradients, [7] defines functions E_i, for red, green and blue, as:

$$E_i = \frac{1}{\sqrt{1 + D(P_5)^2 + D(P_i)^2}}, \qquad (4)$$

where D is the difference in the direction of P_i. As two examples, $E_6 = (1 + D_x(P_5)^2 + D_x(P_6)^2)^{-1/2}$ and $E_3 = (1 + D_{xd}(P_5)^2 + D_{xd}(P_3)^2)^{-1/2}$.

Next, Kimmel [7] observes that if the ratio blue to green is constant within an object, so must be the ratio green to blue. If the ratio green to blue is constant within an object, then locally the ratio green to blue must be the average of the neighboring ratios. The green values are then adjusted appropriately. Of course, this offsets the original blue to green ratio. We go back and fourth three times correcting for the ratio rule of both blue and red. Our implementation is very similar to Kimmel's [7], but we introduce a slight modification. In correcting for the green values, instead of averaging over the entire neighboring pixels, we average only over the pixels where the greens are known.

- Repeat three times:
- Correct the Green values to fit the green over blue ratio test

$$G_5^B = B_5 \frac{E_2 \frac{G_2}{B_2} + E_4 \frac{G_4}{B_4} + E_6 \frac{G_6}{B_6} + E_8 \frac{G_8}{B_8}}{E_2 + E_4 + E_6 + E_8} \qquad (5)$$

$$G_5^R = R_5 \frac{E_2 \frac{G_2}{R_2} + E_4 \frac{G_4}{R_4} + E_6 \frac{G_6}{R_6} + E_8 \frac{G_8}{R_8}}{E_2 + E_4 + E_6 + E_8} \quad (6)$$

and average between the Blue and Red interpolation results

$$G_5 = \frac{G_5^B + G_5^R}{2}$$

- Correct the Blue and Red values via the ratio rule

$$B_5 = G_5 \frac{\sum E_i \frac{B_i}{G_i}}{\sum E_i}, \text{with } i \neq 5 \quad (7)$$

$$R_5 = G_5 \frac{\sum E_i \frac{R_i}{G_i}}{\sum E_i}, \text{with } i \neq 5 \quad (8)$$

- End of loop.

The enhancement steps of equations (5)-(8) tend to introduce some slight artifacts that can almost be classified as noise. In [7] this noise it taken out by a smoothing step. For our smoothing part, we have tried a few different smoothing algorithms including the ones proposed in [7] and in [5].

3. RESULTS

Before we take a look at the results, we would like to mention that the images are color and they do not reproduce well when printed, especially if printed black and white. In order to observe the differences in the interpolation methods, we just printed the red channels for all the images. The see the original, color pictures we strongly urge the reader to view them at:
http://www.texas.cornell.edu/demosaic.

In Fig. 3 we have the original image, which we then down-sampled using the Bayer array of Fig. 1. In Fig. 4 we have the reconstruction results using linear interpolation. Serious artifacts can be noticed around the window and by the house siding. An improvement over direct linear interpolation is to average over the ratios blue to green and red to green (Fig. 5).

The artifacts of linear interpolation are somewhat reduced using the algorithm of [2] (Fig. 6). Better still, our implementation of the algorithm found in [7] seems to produce good results. The artifacts around the window and the house siding are significantly reduced (Fig. 7 and Fig. 8).

4. CONCLUSION

A slight modification of the results of [7] seems to produce some of our best results. There is still room for improvement. For example, the smoothing step in [7] is used because artifacts tend to be introduced by equations (5)-(8). Those artifacts are introduced at known locations. In particular, at the red CFA locations, blue tends to equal green and at the blue locations, red tends to equal green. We believe that the smoothing step can be eliminated, if we are careful in how we handle the trouble spots.

Finally, we would to take this opportunity to thank Prof. Ron Kimmel for the many helpful emails and for providing us with the benchmark images. We would also like to thank Hsien-Che Lee, for the many helpful discussions on color imaging.

Fig. 3. Original Image

Fig. 4. Linear Interpolation of Bayer Array

5. REFERENCES

[1] Günter Wyszecki and W. S. Stiles, "Color Science: Concepts and Methods, Quantitative Data and Formulae," *John Wiley and Sons* , (1982).

[2] 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).

Fig. 5. Linear Interpolation of Ratios

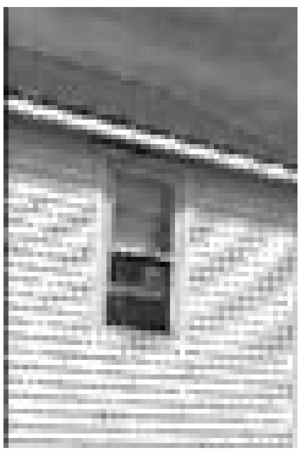

Fig. 6. Cok based interpolation of Bayer Array

[3] D. R. Cok, “Reconstruction of CCD Images Using Template Matching,” *Proc. if IS&T’s Anual Conference/ICPS*, (1994), 380-385.

[4] J.E. Adams, Jr., “Design of Practical Color Filter Array Interpolation Algorithms for Digital Cameras,” *Proceedings of SPIE,* D. Sinha, ed. Vol. 3028, pp. 117-125, SPIE, Bellingham, WA. 1997.

[5] Michael J. Black, Guillermo Sapiro, David H. Marimont, and David Heeger, “Robust Anisotropic Diffusion,” *IEEE Transactions on Image Processing,* Vol. 7, No. 3, (March 1998), pp. 421-432.

[6] Nir Sochen, Ron Kimmel, and Ravi Malldi, “A General Framework for Low Level Vision,” *IEEE Transactions on Image Processing (Special Issue on PDE based Image Processing),* Vol. 7, No. 3 (1998), pp. 310-318.

[7] Ron Kimmel, “Demosaicing: Image Reconstruction from Color CCD Samples,” *IEEE Transactions on Image Processing*, Vol. 8, No. 9, (September 1999), pp. 1221-1228.

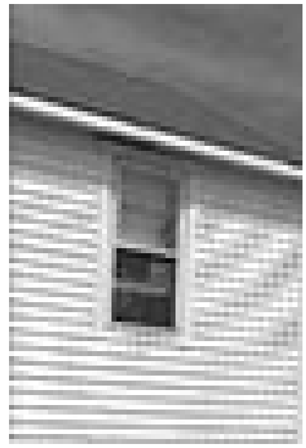

Fig. 7. Kimmel based interpolation of Bayer Array

Fig. 8. Kimmel based interpolation of Bayer Array, followed by the smoothing step of [7] with $\alpha = 0.1$, $\rho = 4$ and a step of 0.05.