Color Filter Array Recovery Using a Threshold-based Variable Number of Gradients

Ed Chang, Shiufun Cheung and Davis Pan

Compaq Computer Corporation, Cambridge Research Laboratory,
Cambridge, Massachusetts, U.S.A.

ABSTRACT

The increase in the popularity of digital cameras over the past few years has provided motivation to improve all elements of the digital photography signal chain. As a contribution towards this common goal, we present a new CFA (Color Filter Array) recovery algorithm, which recovers full-color images from single-detector digital color cameras more accurately than previously published techniques.

This CFA recovery algorithm uses a threshold-based variable number of gradients. In order to recover missing color information at each pixel, we measure the gradients in eight directions based on a 5×5 neighborhood surrounding that pixel. Each gradient value is defined as a linear combination of the absolute differences of the like-colored pixels in this neighborhood. We then consider the entire set of eight gradients to determine a threshold of acceptable gradients. For all of the gradients that pass the threshold test, we use color components from the corresponding areas of the 5×5 neighborhood to determine the missing color values.

We test our CFA recovery algorithm against bilinear interpolation and a single-gradient method. Using a set of standard test images, we show that our CFA recovery algorithm reduces the MSE by over 50% compared to conventional color recovery algorithms. In addition, the resolution test we developed also show that the new CFA recovery algorithm increases the resolution by over 15%. The subjective qualities of test images recovered using the new algorithm also show noticeable improvement.

Keywords: Digital Camera, Color Filter Array, Bayer Pattern, Image Processing

1. INTRODUCTION

We present a method to improve the quality of color images from single-detector imaging systems. Digital color cameras typically use a single image detector to lower costs. Color imaging with a single detector requires the use of a Color Filter Array (CFA) which covers the detector array. In this arrangement each pixel in the detector samples the intensity of just one of the three-color separations. One possibility is to use filters of the primary colors red, green, and blue. Another is to use their complements: cyan, magenta, and yellow. The recovery of full-color images from a CFA-based detector requires a method of calculating values of the other two color separations at each pixel. We call these methods CFA recovery algorithms. Several simple CFA recovery algorithms have been proposed [2–5]. Rising speeds of both general-purpose microprocessors and embedded processors justify more computationally complex methods for the recovery of full-color images. Therefore we have a more computationally complex CFA recovery algorithm, which results in more accurate color recovery for most natural images.

This paper starts by providing a more detailed look at a specific Color Filter Array—the Bayer CFA. Next some previous approaches to CFA recovery are discussed. This is followed by a description of our new CFA recovery algorithm. Finally, we compare our new algorithm to two previous approaches, using mean-square error, our own resolution test and subjective evaluation.

2. BACKGROUND

2.1. Color Filter Arrays

In a single-detector camera, varying intensities of electromagnetic radiation are measured at an image plane composed of a rectangular array of sensor elements. If every element in the array receives the full visible light spectrum, a monochromatic image is produced. As mentioned earlier, construction of a color image requires a color filter array (CFA) be placed between the lens and the sensors at the image plane. A CFA has one color filter element for each sensor. [1]

Part of the IS&T/SPIE Conference on Sensors, Cameras, and Applications
for Digital Photography • San Jose, California • January 1999
SPIE Vol. 3650 • 0277-786X/99/$10.00

Many different CFA configurations have been proposed. One of the most popular is the Bayer pattern, which uses the three primary colors, red, green, and blue (RGB), for the filter elements. It is known that a combination of three primary colors is sufficient to reproduce most colors for visual perception. However, because each sensor can only measure the intensity of a single color, interpolation is required to recover a full color image from the sparsely sampled color values.

In most CFA configurations, one of the colors is sampled more frequently. In the Bayer CFA, the image plane is partitioned in blocks of four sensors with the following pattern:

G R
B G

There are two green sensors on one diagonal, and one red and one blue sensor on the other. The dominant color, green, has, among the three colors, the maximum sensitivity in the human eye. Therefore green is used to represent high-frequency detail.

2.2. Previous approaches to CFA recovery

Recovery of a full-color image can be achieved by the use of simple bilinear interpolation on the measured intensity of the green sensors to determine the luminance at every image pixel [2]. After the luminance has been determined, the full RGB color values at each location in the image can be interpolated from the chrominance values of neighboring sensors. However, a disadvantage of simple bilinear interpolation is that it is prone to color edge artifacts.

One way to minimize this problem is to use adaptive color plane interpolation. In this method proposed by Laroche *et al* [3], the gradients of luminance values are determined in the horizontal and vertical directions. The gradient values in these two directions are compared and a single preferred direction is selected. The other color components at each pixel location are interpolated from nearby sensors located in the preferred direction.

Hamilton *et al* [4] proposed a more sophisticated adaptive algorithm. In their method, the single preferred direction is selected by using a classifier, which is computed by adding LaPlacian second-order values to the gradient values. Then, as in the previous method, the preferred direction is used to perform the interpolation.

There are problems with choosing only a single direction, either horizontal or vertical, for interpolation. In real images, there may be edges that are not predominantly vertical or horizontal. In addition, the image may have many fine-detailed features in which a specific single directional orientation of the gradient does not apply. Therefore, it is desirable to provide a method for interpolating additional color values at locations of the image that do not have a single directional gradient bias. Recently, Wu *et al* [5] partially described an algorithm that uses the weighted average of neighboring like-colored pixels to recover the color components. The algorithm we developed is, in some respects, an extension of this idea.

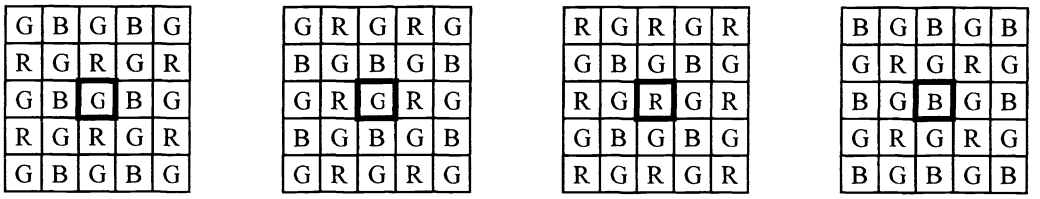
3. NEW METHOD

3.1. Outline and setup

We will now describe our approach to CFA recovery. Although this description is specifically for recovery from the Bayer CFA, the basic ideas of the algorithm can be applied to other CFA configurations.

For each pixel, we recover the missing color components as follows. First, a set of gradients is determined from the color values in the 5×5 neighborhood centered at the pixel under consideration. Each gradient corresponds to a different direction. Second, for each set of gradients, a threshold value is determined, and the threshold is used to select a subset of gradients. Low-valued gradients indicate pixels having similar color values whereas high-valued gradients would be expected in regions of the image where there are many fine details or sharp edges. Third, the subset of gradients is used to locate regions of pixels that are most like the pixel under consideration. Note that the pixels in the identified regions can lie in more than one direction from the pixel under consideration, in contrast to previous work where color values located in only a single direction are used for interpolation. The pixels in the regions are then weighted and summed to determine the average difference between the color of the actual measured center pixel value and the missing color.

Because the Bayer color filter array is periodic with period 2 in both vertical and horizontal directions, there are four separate cases to consider, as shown in Figure 1. The pixel under consideration is in the center of the neighborhood.



(a) Green center, red above (b) Green center, blue above (c) Red center (d) Blue center

Figure 1: Four types of pixels to have color recovery performed

The blue and red pixel arrays have the same sampling pattern, offset by 1 pixel in each dimension. Therefore we can use the same algorithm to recover colors for cases (a) and (b), the two cases in which we recover colors for a green pixel in the center. We can also use one algorithm to recover colors for the other two cases, (c) for a red pixel in the center, and (d) for a blue pixel in the center. In section 3.2 we consider cases (a) and (b), and in section 3.3 we consider (c) and (d).

3.2. Green center

For the cases in which we want to recover the red and blue values at a location where only the green value is known, we perform the following operation. Without loss of generality, we assume case (a) from Figure 1, in which a red pixel is above the center green pixel. Case (b) with a blue pixel above is analogous and solved the same way. For ease of reference, we label the pixels in Figure 2 below.

g1	b1	g2	b2	g3
r1	g4	r2	g5	r3
g6	b3	g7	b4	g8
r4	g9	r5	g10	r6
g11	b5	g12	b6	g13

Figure 2: Green center, red above, pixels numbered for reference

We now describe the steps to recover the red and blue values at the center pixel location. The first step is to determine a set of gradients that will indicate the similarity between the center pixel and the surrounding pixels in each direction. We consider a set of eight gradients in the following eight compass-point directions: N, NE, E, SE, S, SW, W, and NW. In order to determine the gradients, the absolute values of the differences between pairs of similar-colored pixels are weighted and summed as follows:

$$\text{Gradient N} = |r2-r5| + |g2-g7| + |g4-g9|/2 + |g5-g10|/2 + |b1-b3|/2 + |b2-b4|/2, \quad (1)$$

$$\text{Gradient E} = |b4-b3| + |g8-g7| + |g5-g4|/2 + |g10-g9|/2 + |r3-r2|/2 + |r6-r5|/2, \quad (2)$$

$$\text{Gradient S} = |r5-r2| + |g12-g7| + |g9-g4|/2 + |g10-g5|/2 + |b5-b3|/2 + |b6-b4|/2, \quad (3)$$

$$\text{Gradient W} = |b3-b4| + |g6-g7| + |g4-g5|/2 + |g9-g10|/2 + |r1-r2|/2 + |r4-r5|/2, \quad (4)$$

$$\text{Gradient NE} = |g5-g9| + |g3-g7| + |b2-b3| + |r3-r5|, \quad (5)$$

$$\text{Gradient SE} = |g10-g4| + |g13-g7| + |b6-b3| + |r6-r2|, \quad (6)$$

$$\text{Gradient NW} = |g4-g10| + |g1-g7| + |b1-b4| + |r1-r5|, \quad (7)$$

$$\text{Gradient SW} = |g9-g5| + |g11-g7| + |b5-b4| + |r4-r2|. \quad (8)$$

For the purpose of further discussion, we will use, as an example, the numerical values in Table 1 for our eight gradients:

Gradient	*N*	*E*	*S*	*W*	*NE*	*SE*	*NW*	*SW*
Value	12	13	7	8	4	7	12	14

Table 1: An example set of possible gradient values.

The second step is to determine a threshold, below which a subset of gradients is selected. In our implementation, the threshold T is defined as.

$$T = (k_1\ Min) + [k_2\ (Max + Min)]. \tag{9}$$

where *Max* is the maximum gradient value in the set, and *Min* is the minimum gradient value.

The first threshold component, $k_1\ Min$, accounts for the case in which the gradients are all very similar, so that we wish to include all of them by setting a threshold that exceeds them. Therefore k_1 is specified to be greater than 1. We have found empirically that a value of $k_1 = 1.5$ gives good results.

The second threshold component, $k_2\ (Max + Min)$, accounts for the case in which there is a significant difference between the maximum and minimum gradient values. In that case, we wish to include the smaller gradients and exclude the larger ones. Therefore we use k_2 to set a cutoff, and we have found empirically that a value of $k_2 = 0.5$ gives good results.

Based on the threshold value T, a subset of gradients is selected such that all gradients in the subset are less than T. Using the numerical values in the example from Table 1, *Max* = 14, *Min* = 4, and T is determined to be 11. Therefore the selected subset of gradients consists of those with value less than 11.

$$\text{Gradient subset} = \{S = 7, W = 8, NE = 4, SE = 7\}. \tag{10}$$

The third step is to locate pixels in the regions corresponding to the subset of gradients and to use those pixel values to determine a color difference between the known color value of the center pixel and the color values to be recovered. In the example above, we denote the average green, blue, and red values in the gradient subset regions as *G, B,* and *R,* determined as follows:

	G	*B*	*R*
S	(g12 + g7) / 2	(b3 +b4 + b5 + b6) / 2	r5
W	(g6 +g7) / 2	b3	(r1 + r2 + r4 + r5) / 4
NE	g5	(b2 + b4) / 2	(r2 + r3) / 2
SE	g10	(b4 + b6) / 2	(r5 + r6) / 2

Table 2: Example of average color component calculations for given gradients.

The above *G, B,* and *R* values are summed to produce G_{sum}, B_{sum}, and R_{sum}. Then the difference of the two sums, $B_{sum} - G_{sum}$, is divided by the number of gradients in the threshold subset to produce a normalized color difference, BG_{diff}. This difference is added to the pixel value under consideration, g7, to produce an estimate for the missing blue value. Likewise the difference, $R_{sum} - G_{sum}$, is divided by the number of gradients to produce a normalized color difference, RG_{diff}, which is added to g7 to produce an estimate for the missing red value. In our example these would be as follows.

$$b7 = g7 + BG_{diff} = g7 + (B_{sum} - G_{sum})/4, \tag{11}$$

$$r7 = g7 + RG_{diff} = g7 + (R_{sum} - G_{sum})/4, \tag{12}$$

where b7 and r7 are the blue and red pixel values at the center location.

It should be apparent that other formulations can be used to determine the threshold values used for selecting interpolation directions. However, unlike previous work in which gradient values are found from a *single* direction, our method performs interpolation based on a region defined by *multiple* directions.

3.3. Red/blue center

For the cases in which we want to recover the green and blue values from a location where only the red value is known, we perform a similar operation. This corresponds to case (c) from Figure 1. Note that case (d) for recovering the green and red

values at a location where only the blue pixel is known is done the same way and will therefore not be shown explicitly. For ease of reference, we label the pixels as in Figure 3.

r1	g1	r2	g2	r3
g3	b1	g4	b2	g5
r4	g6	r5	g7	r6
g8	b3	g9	b4	g10
r7	g11	r8	g12	r9

Figure 3: Red center. The pixel values are numbered for reference.

In order to recover the green and blue values of the center location, we follow the same three steps as described in the previous section. The first step is to calculate a set of eight gradients.

$$\text{Gradient N} = |g4-g9| + |r2-r5| + |b1-b3|/2 + |b2-b4|/2 + |g1-g6|/2 + |g2-g7|/2, \quad (13)$$

$$\text{Gradient E} = |g7-g6| + |r6-r5| + |b2-b1|/2 + |b4-b3|/2 + |g5-g4|/2 + |g10-g9|/2, \quad (14)$$

$$\text{Gradient S} = |g9-g4| + |r8-r5| + |b3-b1|/2 + |b4-b2|/2 + |g11-g6|/2 + |g12-g7|/2, \quad (15)$$

$$\text{Gradient W} = |g6-g7| + |r4-r5| + |b1-b2|/2 + |b3-b4|/2 + |g3-g4|/2 + |g8-g9|/2, \quad (16)$$

$$\text{Gradient NE} = |b2-b3| + |r3-r5| + |g4-g6|/2 + |g7-g9|/2 + |g2-g4|/2 + |g5-g7|/2, \quad (17)$$

$$\text{Gradient SE} = |b4-b1| + |r9-r5| + |g7-g4|/2 + |g9-g6|/2 + |g10-g7|/2 + |g12-g9|/2, \quad (18)$$

$$\text{Gradient NW} = |b1-b4| + |r1-r5| + |g4-g7|/2 + |g6-g9|/2 + |g1-g4|/2 + |g3-g6|/2, \quad (19)$$

$$\text{Gradient SW} = |b3-b2| + |r7-r5| + |g6-g4|/2 + |g9-g7|/2 + |g8-g6|/2 + |g11-g9|/2. \quad (20)$$

The second step is to determine a threshold and select a subset of gradients. We use the same threshold as given in the previous section:

$$T = (k_1\ Min) + [k_2\ (Max + Min)], \quad (21)$$

where $k_1 = 1.5$ and $k_2 = 0.5$ are found to give good results.

Using the threshold value T, a subset of gradients is selected such that all gradients in the subset are less than T.

The third step is to locate pixels in the regions corresponding to the subset of gradients and to use those pixels to determine a color difference between the center pixel color and the color to be recovered. As in the previous section, we determine the average green, blue, and red values in the gradient subset regions and sum them to produce G_{sum}, B_{sum}, and R_{sum}. Then the difference of the two sums, G_{sum} - R_{sum}, is divided by the number of gradients in the threshold subset to produce a normalized color difference, GR_{diff}. This difference is added to the pixel value under consideration, r5, to produce an estimate for the missing green value. Likewise the difference, B_{sum} - R_{sum}, is divided by the number of gradients to produce a normalized color difference, BR_{diff}, which is added to r5 to produce an estimate for the missing blue value.

4. METRICS

We consider three metrics of measuring image quality: MSE (Mean Square Error), subjective quality, and resolution. The MSE of a the CFA-recovered image is found by averaging the squared differences between all of the color components of each pixel in that image with the corresponding pixel color components in the original image. Because of its simplicity, it is the most common method of objectively measuring image quality given a reference image.

For a subjective evaluation, actual pictures must be viewed. We show images generated by our CFA recovery algorithm as well as those of past methods for an accurate side-by-side comparison.

Finally, we developed our own method to measure resolution as follows. We wrote a resolution chart image in PostScript as shown in Figure 4. Each wedge is 2 degrees wide, so the black and white pattern has a period of 4 degrees. There are semi-circles drawn on the pattern at one-inch intervals to estimate the radial distance at which the wedges blur together. The semi-

circles occupy the arc from 45 degrees to 225 degrees, leaving the lower right half of the image unmarked for our resolution measurement program.

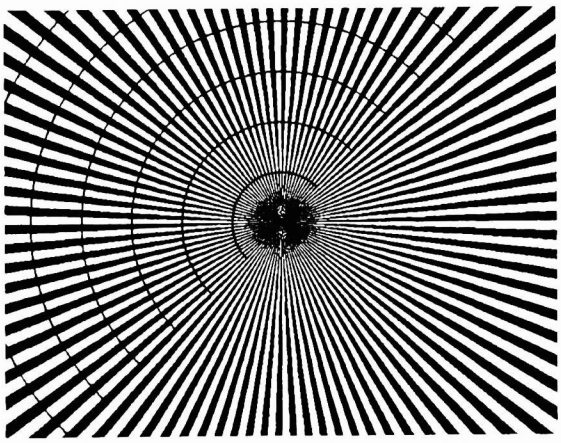


Figure 4: Resolution chart

After converting the chart to a bitmap image file, we filter it with an impulse response to simulate optical blur in a digital camera system. In order to find vertical resolution, we consider a set of adjacent wedges centered on the horizontal midline of the image and opening toward the right. For a given horizontal distance X from the center, we examine all the pixels in the wedge set at that distance and find the average value, **mean**(X). Then we define the difference, **diff**(X,y), to be maximum absolute difference between the pixels and **mean**(X). Finally we define the contrast **c**(X) to be the minimum of the local maxima between the zero crossings of **diff**(X,y). We normalize this contrast function by dividing it by the mean, and we expect it to be a monotonically increasing function of X. By setting a minimum fractional threshold for the contrast, we can find the corresponding distance x, which is in turn proportional to the number of pixels per wedge. Since the focus of our technique is to find the contrast change as a function of resolution, we fix the contrast threshold at an arbitrary value of 10% of maximum contrast for all tests [1]. We assume that at a given x, the wedge width is constant. Then the number of pixels per wedge is equal to the number of pixels per horizontal line, and the reciprocal is the resolution as measured in lines per pixel. The theoretical upper limit of this resolution is one line per pixel. Using the same procedure, we can measure resolution in the horizontal or any other direction by considering the appropriate set of wedges.

5. RESULTS

We first use the resolution measurement test of the previous section to compare our variable-gradient CFA recovery algorithm against the bilinear and single-gradient algorithms. The results are shown in Table 3. The single-gradient method improves on the bilinear by 18%, and the variable-gradient method improves on bilinear by 31%.

Method	*bilinear*	*single-gradient*	*variable-gradient*
resolution	0.55	0.65	0.72

Table 3: Resolution of CFA recovery methods

For real-life images, we choose a set of ten stock images from a Kodak PhotoCD of standard test images. After applying CFA filtering, we compare the MSE results of applying the three methods of recovery, as shown in Figure 5. As seen, the single-gradient method reduces the MSE by 10–30% as compared to bilinear, but the variable-gradient method reduces the MSE by 70–80%.

Finally, we show a real picture comparison using the motocross image. The sharp color edges in this picture present a difficult challenge to CFA recovery algorithms. In Figure 6 we show the full reconstructed images, and the corresponding 50×50 sections, magnified by a factor of 5 to expose details. The magnified sections show clear color aliasing at strong edges though our variable-gradient method results in the least amount of this error.

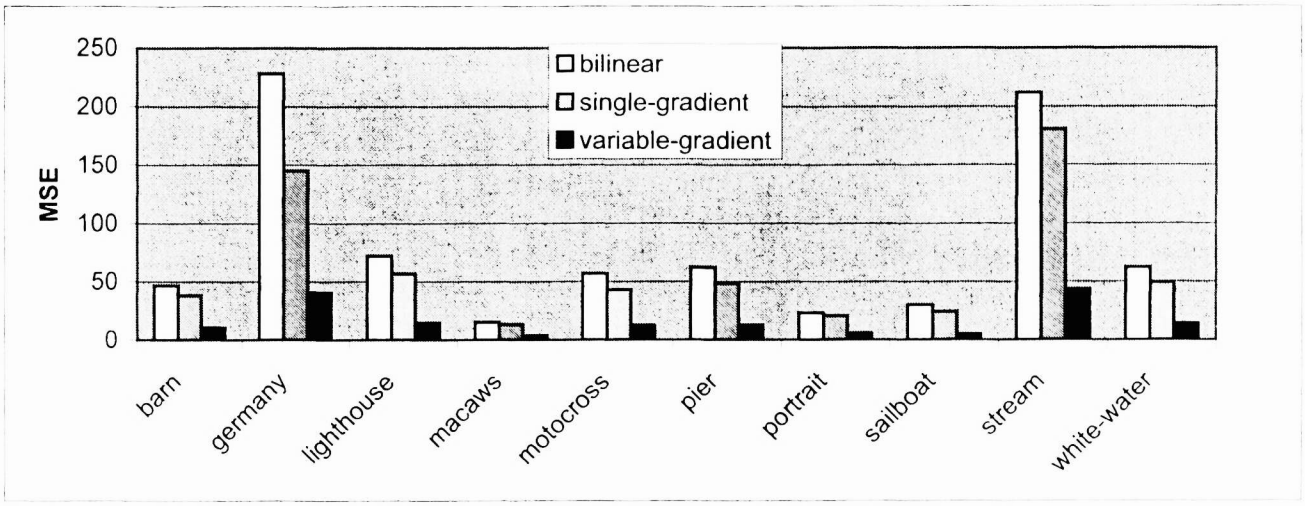


Figure 5: MSE of CFA recovery methods for 10 images

6. CONCLUSION

We have presented a new algorithm for the recovery of color images from a single-detector color digital camera employing a color filter array. This adaptive algorithm extends previous color recovery methods in two major ways. First: a larger area and more directions are searched for similar pixel values, and second, pixel values from all relevant directions are used to compute the missing color values. Our color recovery algorithm performs significantly better than the previous methods in both objective and subjective image quality measures.

REFERENCES

1. Holst, G.C., *CCD Arrays, Cameras, and Displays*, JCD Publishing, Winter Park, FL, 1996
2. Cok, D. R., Signal processing method and apparatus for producing interpolated chrominance values in a sampled color image signal. US patent 4642678, February. 10, 1987.
3. Laroche; C. A. and Prescott; Mark A., Apparatus and method for adaptively interpolating a full color image utilizing chrominance gradients. US patent 05373322, December 13, 1994.
4. Hamilton, Jr., J.F. and Adams, Jr.; J. E., *Adaptive color plan interpolation in single sensor color electronic camera*. US patent 5629734, May 13, 1997.
5. Wu, X., Choi, W. K., and Bao, P., "Color restoration from digital camera data by pattern matching," *Color Imaging: Device-Independent Color, Color Hard Copy, and Graphic Arts II*, Proceedings of SPIE, vol. 3018, pp.12–17, February 1997.

Figure 6: Results from applying different CFA recovery methods to the motocross image: (a) simple bilinear interpolation, (b) single-gradient method, and (c) our variable-gradient method. Corresponding 50×50 enlarged portions are shown on the right.