Color Interpolation for Single CCD Color Camera

Yi-Ming Wu, Chiou-Shann Fuh, and Jui-Pin Hsu
Department of Computer Science and Information Engineering,
National Taiwan University, Taipei, Taiwan
Email: r88036@csie.ntu.edu.tw; fuh@csie.ntu.edu.tw; d92004@csie.ntu.edu.tw

Abstract – This paper addresses the problems of blurred-edge effects and color alias effects in the color interpolation for single charge-coupled device (CCD) color camera, and proposes a combinational method to reduce the problems. We will first introduce the background, and review some traditional interpolation methods. Then we will explain our proposed methods as well as the results. Conclusions and future works will be addressed at the last.

Index Terms – Color interpolation, edge detection, color alias effects, blurred edge effects.

I. Introduction

Presently, the digital still cameras (DSCs) and PC cameras are popular with consumers as a device to input digital images easily. Size reduction and improvement of the image quality are the current trends in developing DSCs. In order to reduce the cost and size, most DSCs use a single charge-coupled device (CCD), instead of using three CCDs, to acquire color image.

However, the single CCD does not provide sufficient color resolutions. The solution for most DSC designers is to cover the sensor's surface with a mosaic of colored filters. This kind of filters is called a color filter array (CFA). An example is Bayer pattern (Fig. 1), a popular arrangement of CFA. Since there is only one color array element in each pixel in CFA, the other two missing color elements must be estimated. Thus, the recovery of full color images from CFA-based sensors requires a process of estimating values of missing color elements at each pixel by its adjacent pixels. This process is commonly called color interpolation or color demosaicking.

We will explain our proposed interpolation methods using Bayer pattern. It should be easily applied to other patterns.

II. Background: Some Commonly Used Interpolation Techniques

Two commonly used interpolation techniques are reviewed here:

1. Bilinear interpolation: Use a linear plane with least square error (against the known neighbor pixels) to interpolate a missing value.
2. Cubic B-spline interpolation [1]: Use a higher-order polynomial sphere with least square error (against the known neighbor pixels) to interpolate a missing value.

These two traditional techniques shares the same side effects: the blurred edge effects and the color alias effects.

III. PROPOSED COLOR INTERPOLATION TECHNIQUES

As we all know, edge-sensitive interpolators can reduce the edge-blurred effects, while color correlation interpolators can reduce the color alias effects. To solve both side effects, we combined the two to get advantages from both.

We will first introduce two edge-sensitive interpolators and the concept of color correlation. Then we will present a method to combine the two.

For convenience, we will use $G_{x,y}$, $R_{x,y}$, and $B_{x,y}$ to denote the pixel values of green, red, and blue at the position *(x,y)* respectively. Note that in Bayer pattern, only one color value exists in a fixed location *(x,y)*. The other two colors are to be interpolated.

3.1 AN EDGE-SENSITIVE INTERPOLATOR: BILINEAR INTERPOLATOR WITH EDGE DETECTOR

We use the gradient edge detector and Laplacian edge detector to determine if a horizontal or vertical edge exists. The horizontal edge response $\Delta\tilde{H}$ and the vertical edge response $\Delta\tilde{V}$ can be expressed as:

$$\Delta\tilde{H} = \left|G_{x+1,y} - G_{x-1,y}\right| + \left|2R_{x,y} - R_{x-1,y} - R_{x+1,y}\right|$$

$$\Delta\tilde{V} = \left|G_{x,y+1} - G_{x,y-1}\right| + \left|2R_{x,y} - R_{x,y-1} - R_{x,y+1}\right|$$

Now, we interpolate the missing pixel by:

(Case 1) $\Delta\tilde{H}$ > predefined threshold, and $\Delta\tilde{V} \le$ predefined threshold

A vertical edge is detected, and the interpolation along the horizontal direction is not desired. Thus we can do bilinear interpolation along the vertical direction only (two-way average).

(Case 2) $\Delta\tilde{H} \le$ predefined threshold, and $\Delta\tilde{V}$ > predefined threshold

Similarly, a horizontal edge is detected, and the interpolation along the vertical direction is not desired. Thus we can do bilinear interpolation along the horizontal direction only (two-way average).

(Case 3) Otherwise

We will do the four-way average interpolation (two-dimensional bilinear interpolation).

3.2 ANOTHER EDGE-SENSITIVE INTERPOLATOR: WEIGHTING-BASED INTERPOLATOR

The previous edge-sensitive interpolator can only detect horizontal or vertical edges. For edges in other directions, it may not work well. Thus a weight-based interpolator [2] comes.

Consider a location *(x, y)* with missing green value, we calculate the eight gradients as follows:

$$DG_x(x+1, x-1) = \frac{G_{x+1,y} - G_{x-1,y}}{2}$$

$$DG_x(x+1, x-3) = \frac{G_{x+1,y} - G_{x-3,y}}{4}$$
$$DG_x(x-1, x+1) = \frac{G_{x-1,y} - G_{x+1,y}}{2}$$
$$DG_x(x-1, x+3) = \frac{G_{x-1,y} - G_{x+3,y}}{4}$$
$$DG_y(y+1, y-1) = \frac{G_{x,y+1} - G_{x,y-1}}{2}$$
$$DG_y(y+1, y-3) = \frac{G_{x,y+1} - G_{x,y-3}}{4}$$
$$DG_y(y-1, y+1) = \frac{G_{x,y-1} - G_{x,y+1}}{2}$$
$$DG_y(y-1, y+3) = \frac{G_{x,y-1} - G_{x,y+3}}{4}$$

Then, we assign the four weights as:

$$w_{x-1,y} = \left(1 + \left(DG_x(x+1, x-1)\right)^2 + \left(DG_x(x+1, x-3)\right)^2\right)^{-\frac{1}{2}}$$
$$w_{x+1,y} = \left(1 + \left(DG_x(x-1, x+1)\right)^2 + \left(DG_x(x-1, x+3)\right)^2\right)^{-\frac{1}{2}}$$
$$w_{x,y-1} = \left(1 + \left(DG_y(y+1, y-1)\right)^2 + \left(DG_y(y+1, y-3)\right)^2\right)^{-\frac{1}{2}}$$
$$w_{x,y+1} = \left(1 + \left(DG_y(y-1, y+1)\right)^2 + \left(DG_y(y-1, y+3)\right)^2\right)^{-\frac{1}{2}}$$

Then, we can interpolate $G_{x,y}$ by

$$G_{x,y} = \frac{G_{x-1,y} \times w_{x-1,y} + G_{x+1,y} \times w_{x+1,y} + G_{x,y-1} \times w_{x,y-1} + G_{x,y+1} \times w_{x,y+1}}{w_{x-1,y} + w_{x+1,y} + w_{x,y-1} + w_{x,y+1}}$$

or, alternatively we can save some computation by simplifying the weights as:

$$w_{x-1,y} = \left(1 + \left|DG_x(x+1, x-1)\right| + \left|DG_x(x+1, x-3)\right|\right)^{-1}$$
$$w_{x+1,y} = \left(1 + \left|DG_x(x-1, x+1)\right| + \left|DG_x(x-1, x+3)\right|\right)^{-1}$$
$$w_{x,y-1} = \left(1 + \left|DG_y(y+1, y-1)\right| + \left|DG_y(y+1, y-3)\right|\right)^{-1}$$
$$w_{x,y+1} = \left(1 + \left|DG_y(y-1, y+1)\right| + \left|DG_y(y-1, y+3)\right|\right)^{-1}$$

For positions with defined green value, the weighting-based interpolations of $R_{x,y}$ and $B_{x,y}$ can be similarly derived.

3.3 Color Correlation

To overcome or reduce color-alias effects, some original interpolation schemes [3, 4] were proposed. Among those schemes, the color correlation is frequently mentioned. Kuno and Sugiura [4] proposed a new interpolation method using discriminated color correlation for digital still cameras. In practice, however, their proposed scheme uses a quotient equation, which sometimes fails to reconstruct stably when the denominator is small.

To avoid the quotient, we adopt another image model developed by Adams [5, 6, 7], with the assumption that green and red, or green and blue values are perfectly correlated to each other within a simple offset in a small region, that is,

$$G_{x,y} = R_{x,y} + K_R$$

$$G_{x,y} = B_{x,y} + K_B$$

where K_R and K_B are called the difference domains of the red channel, and blue channel respectively. In practice, K_R and K_B channels are quite flat.

We will not give an explicit interpolation method here, instead, we will give a composite method using the above two equations directly.

3.4 THE COMPOSITE METHOD

We will focus on the weight-based edge-sensitive interpolator plus the color correlation. Combining the gradient and Laplacian edge-sensitive interpolator plus the color correlation is similar.

Now, the whole task will be to guess the unknown pixel values by interpolation. For each coordination (x, y) with known red value, we defined the four K values as:

$$K_{-1,0} = G_{x-1,y} - (R_{x-2,y} + R_{x,y})/2 \ldots\ldots\ldots (A1)$$

$$K_{1,0} = G_{x+1,y} - (R_{x+2,y} + R_{x,y})/2 \ldots\ldots\ldots (A2)$$

$$K_{0,-1} = G_{x,y-1} - (R_{x,y-2} + R_{x,y})/2 \ldots\ldots\ldots (A3)$$

$$K_{0,1} = G_{x,y+1} - (R_{x,y+2} + R_{x,y})/2 \ldots\ldots\ldots (A4)$$

Note that the rightmost term of equation *(A1)* is a estimation of $R_{x-1,\, y}$, and similarly, the rightmost term of equation *(A2), (A3)*, and *(A4)* are estimations of $R_{x+1,\, y}$, $R_{x,y-1}$, and $R_{x,y+1}$ respectively. Thus these K values are actually estimations of $G_{x-1,y}$-$R_{x-1,y}$, $G_{x+1,y}$-$R_{x+1,y}$, $G_{x,y-1}$-$R_{x,y-1}$, and $G_{x,y+1}$-$R_{x,y+1}$. Note that all four K values are well-defined since all those G and R values on the right-hand-side are known. We further defined:

$$K_{-0,0} = G_{x,\, y} - R_{x,\, y} \ldots\ldots\ldots (A5)$$

All these five K values are said to be the value of R in the difference domain, where R stands either for red pixel or blue pixel, depending on the position.

Equation *(A5)* can be rewritten as:

$$G_{x,\, y} = R_{x,\, y} + K_{0,0} \ldots\ldots\ldots (A6)$$

To do interpolation for the missing green values $G_{x,\, y}$, we first do interpolation of $K_{0,\, 0}$ from the four adjacent K values, using one of the edge aware interpolator. Then, use equation *(A6)* to evaluate $G_{x,\, y}$.

Interpolation of the missing values of other positions can be done similarly. However, the four weights should be assigned using green values only, since green values are more accurate because of the domination (Half of the pixels are green).

IV. RESULTS AND CONCLUSION

4.1 SIMULATION OF COLOR INTERPOLATION

To simulate Bayer pattern, original images are down-sampled, to extract only one color band per pixel, according to Bayer pattern. Figure 2 shows the images before and after down-sampling. Figure 3 shows the original image with the interesting region in a bounded box. Figure 4 through 6 shows some interpolation results.

4.2 PEAK SIGNAL TO NOISE RATIO (PSNR) RESULTS AND COMPARISONS

We now apply the PSNR to measure the similarity between the original images and the image which is down-sampled followed by various interpolation methods. The PSNR value is defined as follows:

$$PSNR(I_o, I_r) = 10 \times \log_{10}\left(\frac{255^2}{MSE(I_o, I_r)}\right) \qquad (4.1)$$

$$MSE(I_o, I_r) = \frac{1}{H \times W} \times \sum_{y=0}^{H-1}\sum_{x=0}^{W-1}\left(I_o(x, y) - I_r(x, y)\right)^2 \qquad (4.2)$$

where I_o is the original image; I_r is the resulted image; *MSE* is the mean square error; *H* is the height of the image; *W* is the width of the image; and 255 is the maximum value which a pixel can have in the 8-bits image.

In our experiments, we take 100 images shown in Figures 7 and 8 as our inputs. We assign a number to every image; Images 001 to 090 are natural scenes in our lives and Images 091 to 100 are artificial paints. We down-sampled these 100 images and then apply various interpolation methods to them. The interpolation methods which will be compared in this experiment are nearest neighbor, bilinear, cubic B-spline, cubic convolution, edge-filtered, weighting-based, edge-filtered color-difference, weighting color-difference, and simplified weighting color-difference. The first four methods are traditional interpolation methods, the next two methods are edge-sensitive interpolation methods, and the last three methods are our proposed composite interpolation schemes. At last, the PSNR values are calculated.

For clearer comparison, we assign a score to every method according to the rank of the PSNR value. We assign score 1 to the minimum PSNR (worst) among all methods, score 2 to the next better place, and increasing scores to the others according to their places, the maximum PSNR (best) will get score 9. After scoring, we summed total scores per method and the result is shown in Table 4 and observed that the weighting color-difference interpolation method has the best score and its simplified version is slightly worse. Summarily, the composite interpolation methods have better scores, the edge-sensitive methods are next, and the traditional interpolation methods are worst. Besides, the weighting-based method is better than edge-filtered method. Among traditional interpolation, the cubic convolution is the best and the nearest neighbor is the worst.

V. Bibliography

[1] D. P. Mitchell and A. N. Netravali, “Reconstruction Filters in Computer Graphics,” *Computer Graphics*, vol. 22, no. 4, pp. 221-228, 1988.

[2] S. Carrato, G. Ramponi, and S. Marsi, “A Simple Edge-Sensitive Image Interpolation Filter,” *Proceedings of International Conference on Image Processing*, vol. 3, pp. 711-714, 1996.

[3] W. T. Freeman, “Median Filter for Reconstructing Missing Color Samples,” United States Patent, 4724395, 1998.

[4] T. Kuno and H. Sugiura, “New Interpolation Method Using Discriminated Color Correlation for Digital Still Cameras,” *IEEE Transactions on Consumer Electronics*, vol. 45, no. 1, pp. 259-267, 1999.

Image No.	Nearest Neighbor	Bilinear	Cubic B-spline	Cubic Convolution	Edge-Filtered	Weighting-Based	Edge-filtered Color-Difference	Weighting Color-Difference	Simplified Weighting Color-Difference
001 ~ 090	90	290	193	414	402	508	658	**791**	703
091 ~ 100	10	35	24	38	51	52	72	83	**85**
001 ~ 100	100	325	217	452	453	560	730	**874**	788

Table 1: The total of the PSNR comparison scores. The bold-faced green text indicates the best score and the red text indicates the worst score.

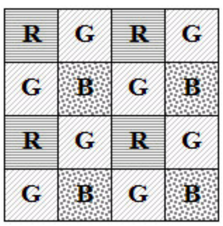

Figure 1: Bayer pattern for CFA.

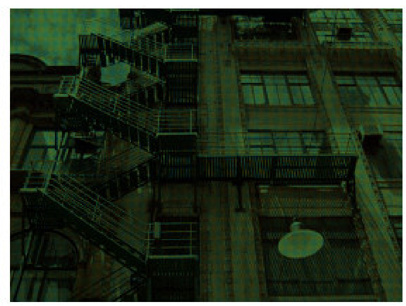

Figure 2: (a) The original image. **(b) The result of the down-sampled (a).**

Figure 3: The original image for testing and the bounding box in the center is the region of interest.

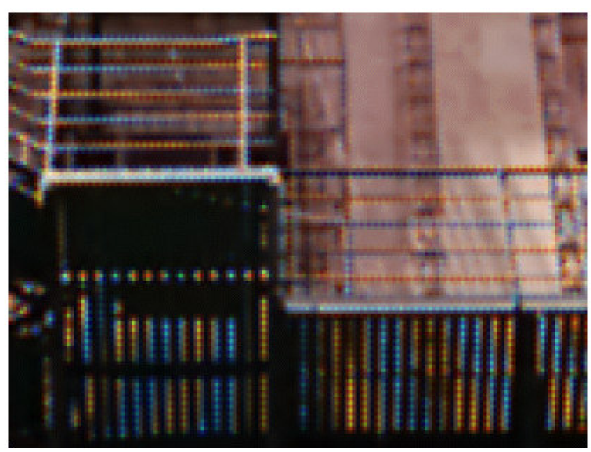

Figure 4: The interpolated image with bilinear interpolation, showing the scale-up image of the bounded box, where the average PSNR is 25.21 dB.

[5] J. E. Adams, Jr., “Design of Practical Color Filter Array Interpolation Algorithms for Digital Cameras,” *Proceedings of SPIE*, vol. 3028, pp. 117-125, 1997.

[6] J. E. Adams, Jr., “Design of Practical Color Filter Array Interpolation Algorithms for Digital Cameras, Part 2,” *Proceedings of International Conference on Image Processing*, vol. 1, pp. 488-492, 1998.

[7] J. Adams, K. Parulski, and K. Spaulding, “Color Processing in Digital Cameras,” *IEEE Micro*, vol. 18, no. 6, pp. 20-30, 1998.

Figure 5: The interpolated image with edge-filtered color-difference interpolation, showing the scale-up image of the bounding box, where the average PSNR is 34.52 dB.

Figure 6: The interpolated image with weighting color-difference interpolation, showing the scale-up image of the bounding box, where the average PSNR is 35.72 dB.

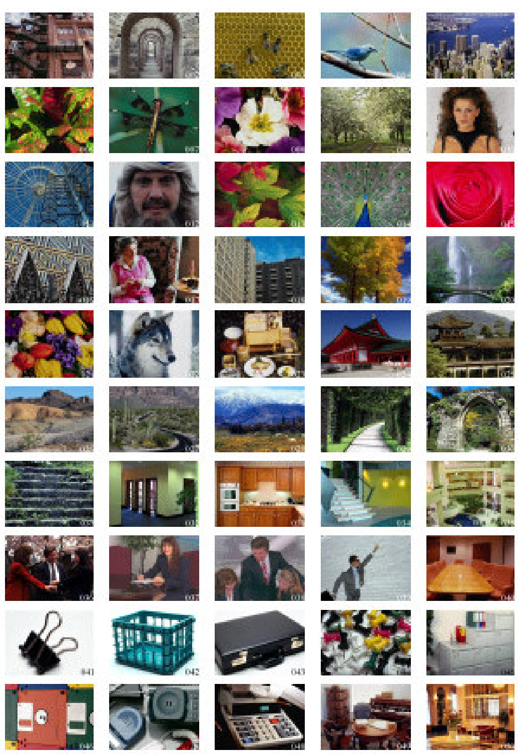

Figure 7: The first 50 testing images (640 x 480 pixels) to be measured, Images 001 to 050.

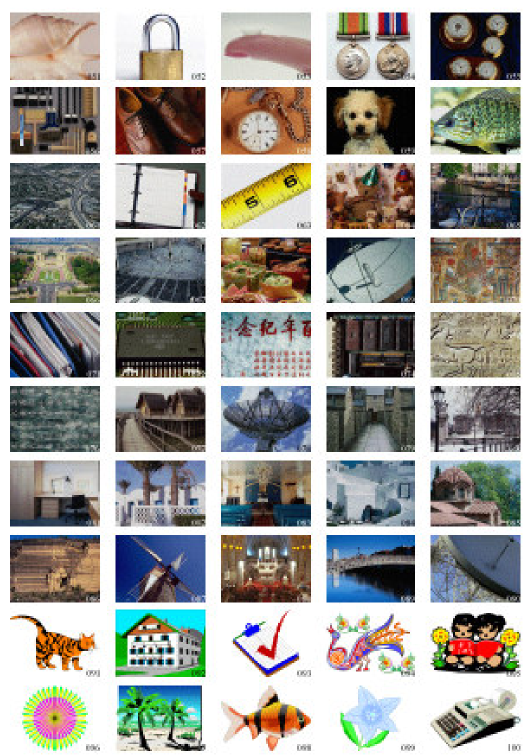

Figure 8: The last 50 testing images (640 x 480 pixels) to be measured, Images 051 to 100.