Research Journal of Applied Sciences, Engineering and Technology 6(18): 3384-3388, 2013
DOI:10.19026/rjaset.6.3652
ISSN: 2040-7459; e-ISSN: 2040-7467

Submitted: January 19, 2013 Accepted: March 02, 2013 Published: October 10, 2013

Research Article

Color Restoration of Monochrome Image Formatted by Y800

[1]Jun Luo, [1]Rui Su and [2]Ying Chen
[1]College of Science, Huazhong Agricultural University, Wuhan, Hubei Province, 430070, China
[2]Hubei Province Fiber Inspection Bureau, Wuhan, Hubei Province, 430060, China

Abstract: In order to restore the color of original monochrome image which obtained from Charge-Coupled Device (CCD) directly, we design a Bayer mode color filter array start with specific pixels to satisfy the imaging condition and then we use bilinear interpolation algorithm to restore the color of original monochrome image which formatted by Y800, however, the hues of adjacent pixels change abruptly by the bilinear interpolation, therefore, we use smooth hue transition interpolation to enhance the edge of original image and finally we identify the ultimate performance of these interpolation algorithms by magnified the interpolated image and use mean square error to compare these algorithms, the results show that the smooth hue transition interpolation is better than bilinear interpolation for color restoration of monochrome image.

Keywords: Bayer color filter array, interpolation algorithm, Y800 data

INTRODUCTION

Usually, we can obtain color image by three-CCD system, or single-CCD camera alternatively. In three-CCD system, each CCD is corresponding to a color value of RGB, so we can obtain a colorful image in this imaging condition, However, the price of this system is so expensive that we can not apply them in reality, so, it is essential for us to use single-CCD color digital camera (Yi *et al*., 2004). In fact, CCD chip itself does not record any color value, so in order to restore the color information of original image, we need to apply a specific color interpolation algorithm to restore the other two color information, such as GB, RG or RB. The solution is to place a Bayer color filter array before the single-CCD chip (Bahadir and John, 2005). Thus, each element of the array stands for the pixel corresponding to the one color value of RGB, because the pixel can only obtain one color value, such as R, G or B. So, we need to restore the other two color value of every pixel. There are many image interpolation algorithms to restore the color information, such as the nearest neighbor interpolation algorithm, the bilinear interpolation algorithm, the smooth hue transition interpolation algorithm and so on (Sakamoto, 1998).

In our study, we will introduce the imaging model and Bayer color filter matrix for the first, then give some introduction of interpolation algorithms and identify the ultimate performance of these algorithms by several experiments finally.

BAYER IMAGING MODEL AND INTERPOLATION ALGORITHM

Usually, the Y800 data is obtained directly from the CCD sensor without any modification by the processing

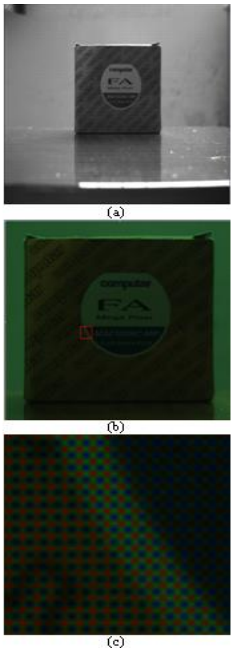


Fig. 1: (a) The Y800 original data of a box, (b) convert the Fig. 1a to a 24-bit image, (c) the enlargement of the small red box of Fig. 1b

Corresponding Author: Jun Luo, College of Science, Huazhong Agricultural University, Wuhan, Hubei, 430070, China

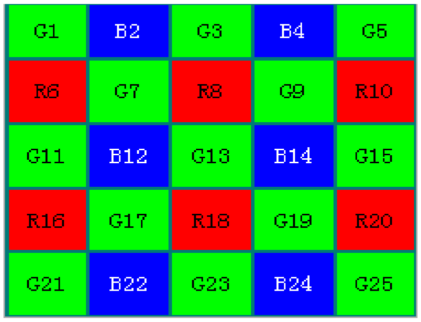
units of camera, which called original monochrome image which can reflect the intensity of special light at every pixel. However, the Y800 data cannot be applied directly because they are not color image in some degree. In this study, we use Imaging-source camera DFK41BU02 made in Germany to obtain the Y800 original monochrome image. As can be seen in Fig. 1a, we take a photo of a box stored as Y800 original data by this camera, which is an 8-bit data file. In this original data, each pixel represents the grayscale value of the whole image sensor, which is one of the R, G or B value and the pixel doesn't contain the other two color components, such as the value of GB, RG or RB (Wen and Sheng, 2006). In Fig. 1b, we convert the Fig. 1a to a 24-bit image and we can clearly see the enlargement of the small red box of Fig. 1b in 1c.

In order to restore the color information of Fig. 1a, we need to calculate the other two color component values. As we all known, many cameras are manufactured by Bayer model, so in accord with the Bayer imaging model, we can restore the color information by two steps. For the first, we should design a Bayer color filter array which start with two particular pixels for original image data, secondly, it should choose a suitable and efficient color interpolation algorithm for the Bayer color filter array to restore the other two color component values of Y800 original data.

We can see a Bayer color filter array start with GB pixels in Fig. 2, this array is placed in front of the CCD chip and each pixel only stand for one color value of RGB, so the Y800 data is not a color image which reflects the intensity of special light for every pixel. In Fig. 2, R, G and B stand for red, green and blue respectively, for example: the pixel named G1 represents the green value of the first pixel, B2 represents the blue value of the second pixel and R6 stands for the red value of the sixth pixel. Its starting color value determines the starting value of the original data which the base value. Color filter array is different, the starting model is different and so algorithm needs to make adjustments. Usually, there are four Bayer color filter arrays start with GB, BG, RG or GR pixels and different start model need different interpolation algorithms, but only one model is correct.

In this study, we use GB model because imaging-source camera is arranged by this imaging model, which can be seen in Fig. 2.

As can be seen in Fig. 2, the Bayer color filter array just maintain one color value, such as R, G or B, the other two color values of each pixel are removed by Bayer filter, so we need to restore the other two color information of each pixel. In this study, we use bilinear interpolation algorithm and smooth tone transition interpolation algorithm for the Y800 original data.



G1	B2	G3	B4	G5
R6	G7	R8	G9	R10
G11	B12	G13	B14	G15
R16	G17	R18	G19	R20
G21	B22	G23	B24	G25

Fig. 2: The bayer color filter matrix models starting with GB pixels

Bilinear interpolation algorithm for Y800 data: According to the structure of Bayer color filter array depicted in Fig. 2, we use bilinear interpolation algorithm to restore the color information of Y800 original data. The algorithm can be designed as follows:

Interpolation of green pixels: The average of the upper, lower, left and right pixel values is assigned as the G value of the interpolated pixel. For example, in Fig. 2, we can obtain the values of G8 at R8 and G12 at B12 by Eq. (1):

$$\begin{cases} G8=(G3+G7+G9+G13)/4 \\ G12=(G7+G11+G13+G17)/4 \end{cases} \tag{1}$$

Interpolation of red/blue pixels at a green position: The average of two adjacent pixel values in corresponding color is assigned to the interpolated pixel. For example, in Fig. 2, we can obtain the values of B7 and R7 at G7 by Eq. (2) and B13 and R13 at G13 by Eq. (3):

$$\begin{cases} B7=(B2+B12)/2 \\ R7=(R6+R8)/2 \end{cases} \tag{2}$$

At G13, it corresponds to the B13 and R13 values for:

$$\begin{cases} B13=(B12+B14)/2 \\ R13=(R8+R18)/2 \end{cases} \tag{3}$$

Interpolation of a red/blue pixel at a blue/red position: The average of four adjacent diagonal pixel values is assigned to the interpolated pixel. For example, in Fig. 2, we can obtain the values of B8 at R8 and R12 at B12 by Eq. (4):

$$\begin{cases} B8=(B2+B4+B12+B14)/4 \\ R12=(R6+R8+R16+R18)/4 \end{cases} \tag{4}$$

Smooth hue transition interpolation algorithm: One of the key objections of the bilinear interpolation is that the hues of adjacent pixels change abruptly and in an unnatural manner. The Bayer CFA pattern, on the other hand, can be thought of as consisting of a luminance channel (the relatively numerous green pixels) and a chrominance channel (the relative sparse red and blue pixels). A scheme can be created to interpolate these channels differently. This algorithm is based on Mondrian and. According to Mondrian model, the color ratio rule of adjacent pixels (m, n) and (i, j) can be defined as Eq. (5):

$$\begin{cases} R_{mn}/G_{mn} = R_{ij}/G_{ij} \\ B_{mn}/G_{mn} = B_{ij}/G_{ij} \end{cases} \tag{5}$$

Interpolation of green pixels: In this algorithm, the method for obtaining green pixel values is the same as bilinear interpolation. Interpolated green pixel values is in front of the interpolate pixel values red and blue.

Interpolation of blue pixels: The idea here is to try to impose a smooth transition in hue from pixel to pixel. To do so, we define the blue "hue value" and red "hue value" as Eq. (6):

$$\begin{cases} \text{blue hue} = B/G \\ \text{red hue} = R/G \end{cases} \tag{6}$$

Considering the interpolation of blue pixel values there are three different cases of blue pixel value interpolations.

In Fig. 2, we can estimate blue pixel values at the green position and adjacent blue pixels on left and right. For example, we can obtain the values of B13 and at G13 by Eq. (7):

$$B13 = G13/2 \times (B12/G12 + B14/G14) \tag{7}$$

In Fig. 2, we can estimate blue pixel value at the green position and adjacent blue pixels on top and bottom. For example, we can obtain the values of B7 and at G7 by Eq. (8):

$$B7 = G7/2 \times (B2/G2 + B12/G12) \tag{8}$$

In Fig. 2, we can estimate blue pixel values at the red position. For example, we can obtain the values of B18 at R18 by Eq. (9):

$$\begin{aligned} &B18 = G18/4 \times (B12/G12 + B14/G14 + \\ &B22/G22 + B24/G24) \end{aligned} \tag{9}$$

Interpolation of red pixels: There are three different cases of red pixel value interpolations:

- Estimating red pixel values in a green position and adjacent to the location of red pixels on the left and right. For example, we can obtain the values of R7 at G7 by Eq. (10):

$$R7 = G7/2 \times (R6/G6 + R8/G8) \tag{10}$$

- Estimating red pixel values in a green position and adjacent to the location of red pixels in the top and bottom. For example, we can obtain the values of R13 at G13 by Eq. (11):

$$R13 = G13/2 \times (R8/G8 + R18/G18) \tag{11}$$

- Estimating red pixel values at the blue position. For example, we can obtain the values of R12 at B12 by Eq. (12):

$$R12 = G12 \Big/ \begin{aligned} &4 \times (R6/G6 + R8/G8 + \\ &R16/G16 + R18/G18) \end{aligned} \tag{12}$$

RESULTS AND DISCUSSION

In the following experiments, we use Imaging-source camera to take a photo of a yellow cup stored as Y800 original data, as can be seen in Fig. 3a. In order to identify the ultimate performance of above interpolation algorithm, we use bilinear and smooth hue transition interpolation algorithm to restore the color information of Y800 original sample.

Bilinear interpolation for four starting pixel model: According to the starting model with GB, BG, GR, RG pixels, we use bilinear interpolation algorithm to restore the color of them respectively, which can be seen in Fig. 3b to e respectively. The results show that the GB model is right for interpolation algorithm and the restoration results of starting model with BG, GR, RG pixels are not correct, because they cannot restore yellow color of the cup.

Smooth hue transition interpolation for four starting pixel model: After the bilinear interpolation for Y800 original image, we use smooth hue transition interpolation algorithm to restore four starting pixel of original data respectively, which can be seen in Fig. 4a to d respectively. The results show that the GB model is still right for interpolation algorithm, the others are not

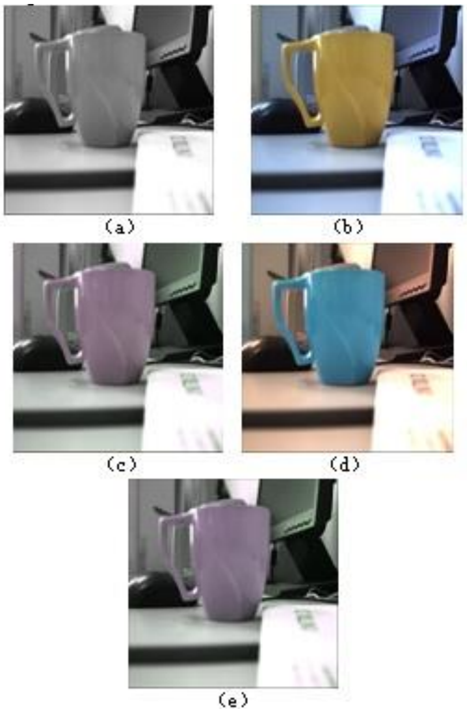


Fig. 3: (a) The Y800 original data of a cup, (b) bilinear interpolation in accordance with GB starting model, (c) bilinear interpolation in accordance with the BG starting model, (d) bilinear interpolation in accordance with the GR starting mode, (e) bilinear interpolation in accordance with the RG starting model

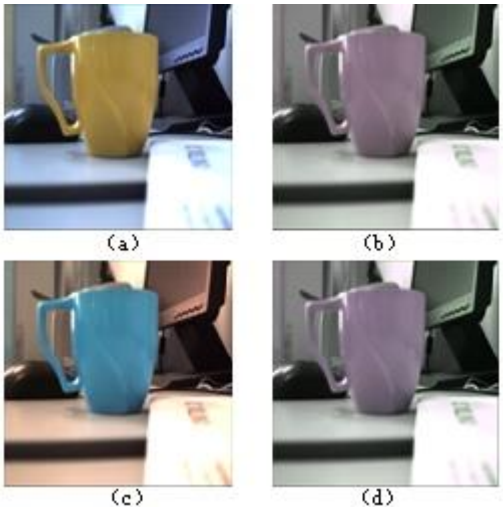
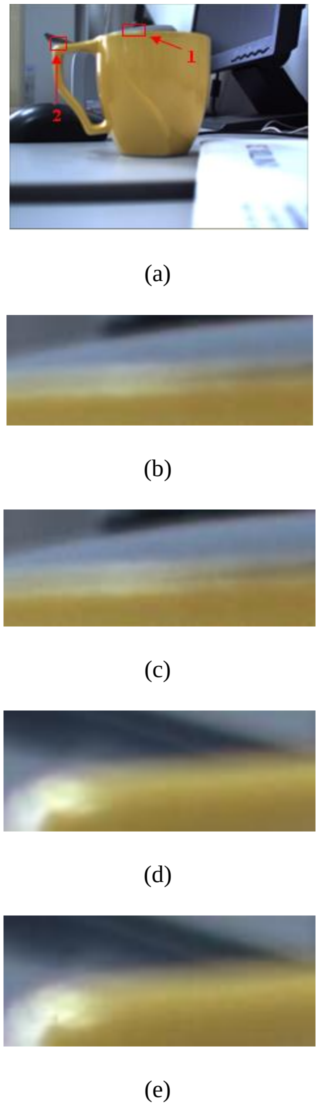


Fig. 4: (a) Smooth hue transition interpolation in accordance with GB starting model, (b) smooth hue transition interpolation in accordance with the BG starting model, (c) smooth hue transition interpolation in accordance with the GR starting mode, (d) smooth hue transition interpolation in accordance with the RG starting mode



(a)

(b)

(c)

(d)

(e)

Fig. 5: (a) Two areas drawn by red line in original yellow cup image, (b) zoom in the area named "1" of restored image by using bilinear interpolation, (c) zoom in the area named "1" of restored image by using smooth hue transition interpolation, (d) zoom in the area named "2" of restored image by using bilinear interpolation, (b) zoom in the area named "2" of restored image by using smooth hue transition interpolation

correct, because they cannot restore yellow color of the cup. Therefore, it is essential for us to pay attention to the starting pixel model of Bayer color filter array.

Compare the magnified image of bilinear and smooth hue transition interpolation algorithm: In accordance with the principle of interpolation algorithms, we analyze the differences of the above two interpolation algorithms. In Fig. 5a, we draw two areas

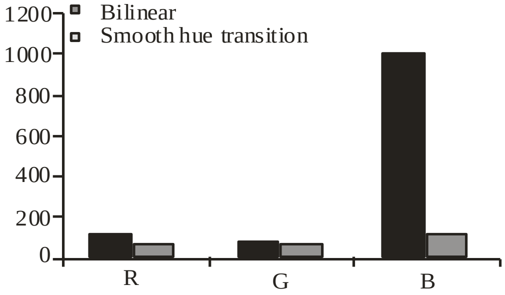


Fig. 6: (a) The mean square error between the original image and the interpolated image for R, G and B channel

in red line named “1” and “2”, then we zoom in the area named “1” and “2” of interpolated image respectively, which can be seen in Fig. 5b to e, the results show that the visual effect of restored image by using smooth hue interpolation algorithm are better than that of using bilinear interpolation algorithm.

Objective quality measure: To compare these algorithms objectively, we adopt Mean Square Error (MSE) as our quality metrics. The MSE between the original image I_o (x, y) and the interpolated image I_r (x, y) for each color channel is defined in Eq. (13), where m×n is the size of the image:

$$MSE = \frac{\sum_{x=1}^{m}\sum_{y=1}^{n}[I_o(x,y) - I_r(x,y)]^2}{m \times n} \tag{13}$$

From Fig. 6, we can see that the restored image by using smooth hue interpolation algorithm is closer to the original image than bilinear interpolation algorithm.

CONCLUSION

In this study, we obtained the original image from single CCD without any color information which called Y800 original data. In order to restore the color information of original image, we design a Bayer color filter array with four starting pixels mode and then we use bilinear interpolation algorithm and smooth hue transition interpolation algorithm to restore the color of original image. The ultimate results show that the smooth hue transition interpolation algorithm is better than bilinear interpolation for color restoration of monochrome image.

ACKNOWLEDGMENT

This study was supported by a grant from the National Natural Science Foundation of China (No. 60777003 and No. 60736010).

REFERENCES

Bahadir, K.G. and G. John, 2005. Demosaicking: Color filter array interpolation. IEEE Signal Proc. Mag., 22(1): 44-54.

Sakamoto, T., 1998. Software pixel interpolation for digital still cameras suitable for a 32-bit MCU. IEEE T. Consum. Electr., 44(4): 1342-1352.

Wen, C.K. and H.W. Sheng, 2006. Designing image processing pipeline for color imaging systems. Proceeding of the IEEE International Symposium on Circuits and Systems (ISCAS 2006), pp: 4679-4682.

Yi, M.W., S.F. Chiou and P.H. Jui, 2004. Color interpolation for single CCD color camera. Proceeding of the 6th IEEE Southwest Symposium on Image Analysis and Interpretation, pp: 90-94.