

COLOUR FILTER ARRAY DEMOSAICKING USING CUBIC SPLINE INTERPOLATION

J. S. Jimmy Li and Sharmil Randhawa

School of Informatics and Engineering, Flinders University, GPO Box 2100, Adelaide 5001, Australia.

ABSTRACT

When a single image sensor is used to capture colour images, only one colour value is acquired at each pixel location. Colour filter array (CFA) demosaicking refers to the estimation of the other two missing colour values in order to produce a full colour image. In this paper, we have shown that cubic spline interpolation is suitable for CFA demosaicking for producing a smooth colour image due to the known fact that it can generate a smooth curve to fit a set of data points. In our approach to obtain a unique solution for cubic spline interpolation, the two required extra conditions were derived based on the correlation between colour planes. Our proposed method, by applying cubic spline interpolation in conjunction with a selector based on edge orientation, has been shown to outperform other existing demosaicking methods.

Index Terms— demosaicking, cubic spline interpolation, colour filter array

1. INTRODUCTION

Colour filter array (CFA) demosaicking is the process of interpolating missing colour values at each pixel location when a single image sensor is used to capture colour images. The most common filter array used is the Bayer [1] pattern as shown in Fig. 1. The green colour is sampled at twice the rate of the red and blue values. This is due to the peak sensitivity of the human visual system which lies in the green spectrum [1].

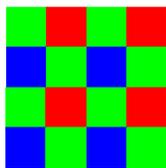


Fig. 1: The Bayer CFA Pattern

Various techniques have been proposed for CFA demosaicking [3,4,5,6,7,9,10,11,12,13,14,15]. However, most of these techniques will produce different degrees of colour artifacts and associated problems. In our approach, we apply the cubic spline interpolation to interpolate the missing green colour values in order to produce a better estimate in an attempt to reduce colour artifacts. The curve

produced by cubic spline interpolation is seamless because it is continuous up to the second order derivative at the data points. These stable and smooth characteristics are desirable because these will lead to the production of a demosaicked output image with smooth colour. However, in order to avoid interpolation across an edge, we use a selector to determine which orientation to apply cubic spline interpolation. For green plane interpolation and based on the Bayer pattern, only the vertical or horizontal directions will be considered because there are no green samples along any diagonal directions at pixel locations where the green values are missing. For the red and blue plane, using first order interpolation will be sufficient to determine their missing values because their sampling rate is only half that of the green plane [10].

2. CUBIC SPLINE INTERPOLATION

Cubic spline interpolation is a piecewise continuous curve, with continuous first and second order derivatives [16]. Given $n+1$ points (x_0, y_0) to (x_n, y_n) , a third degree polynomial, $S_i(x)$, is constructed between each point and is denoted as the cubic spline.

$$S_i(x) = a_i x^3 + b_i x^2 + c_i x + d_i \text{ for } x \in [x_i, x_{i+1}]. \quad (1)$$

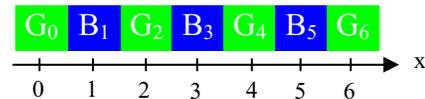


Fig.2: 1D Bayer Pattern

Consider the one-dimensional horizontal Bayer pattern in Fig.2 above. Let G_x and B_x denote the green and blue colour values at position x respectively. In our case, there are four green data points, and hence $n=3$. In order to determine the missing green value at position $x=3$ (i.e. at location B_3), three splines are required as follows:

$$S_0(x) = a_0 x^3 + b_0 x^2 + c_0 x + d_0 \text{ for } x \in [0,2]. \quad (2)$$

$$S_1(x) = a_1 x^3 + b_1 x^2 + c_1 x + d_1 \text{ for } x \in [2,4]. \quad (3)$$

$$S_2(x) = a_2 x^3 + b_2 x^2 + c_2 x + d_2 \text{ for } x \in [4,6]. \quad (4)$$

From (2)-(4), since there are twelve unknown coefficients: $a_0, b_0, c_0, d_0, a_1, b_1, c_1, d_1, a_2, b_2, c_2$ and d_2 , 12 simultaneous equations are required to obtain a unique solution.

For cubic spline interpolation, three continuity conditions have to be met as follows:

First continuity condition:

$$S_{i-1}(x_i) = S_i(x_i) \text{ for } i = 1, 2 \quad (5)$$

Second continuity condition:

$$S'_{i-1}(x_i) = S'_i(x_i) \text{ for } i = 1, 2 \quad (6)$$

Third continuity condition:

$$S''_{i-1}(x_i) = S''_i(x_i) \text{ for } i = 1, 2 \quad (7)$$

Based on these continuity conditions, there are a total of $4n-2$ equations. In our case for $n=3$, there are 10 equations as follows:

By the first continuity condition (5), we have

$$S_0(0) = G_0 = d_0 \quad (8)$$

$$S_0(2) = G_2 = 8a_0 + 4b_0 + 2c_0 + d_0 \quad (9)$$

$$S_1(2) = G_2 = 8a_1 + 4b_1 + 2c_1 + d_1 \quad (10)$$

$$S_1(4) = G_4 = 64a_1 + 16b_1 + 4c_1 + d_1 \quad (11)$$

$$S_2(4) = G_4 = 64a_2 + 16b_2 + 4c_2 + d_2 \quad (12)$$

$$S_2(6) = G_6 = 216a_2 + 36b_2 + 6c_2 + d_2 \quad (13)$$

The first derivatives of the splines are given by:

$$S'_0(x) = 3a_0x^2 + 2b_0x + c_0 \quad (14)$$

$$S'_1(x) = 3a_1x^2 + 2b_1x + c_1 \quad (15)$$

$$S'_2(x) = 3a_2x^2 + 2b_2x + c_2 \quad (16)$$

By the second continuity conditions (6), we have

$$S'_0(2) = S'_1(2) \quad (17)$$

$$\text{Therefore, } 12a_0 + 4b_0 + c_0 - 12a_1 - 4b_1 - c_1 = 0 \quad (18)$$

$$S'_1(4) = S'_2(4) \quad (19)$$

$$\text{Therefore, } 48a_1 + 8b_1 + c_1 - 48a_2 - 8b_2 - c_2 = 0 \quad (20)$$

The second derivatives of the splines are given by:

$$S''_0(x) = 6a_0x + 2b_0 \quad (21)$$

$$S''_1(x) = 6a_1x + 2b_1 \quad (22)$$

$$S''_2(x) = 6a_2x + 2b_2 \quad (23)$$

By the third continuity condition (7), we have

$$S''_0(2) = S''_1(2) \quad (24)$$

$$\text{Therefore, } 12a_0 + 2b_0 - 12a_1 - 2b_1 = 0 \quad (25)$$

$$S''_1(4) = S''_2(4) \quad (26)$$

$$\text{Therefore, } 24a_1 + 2b_1 - 24a_2 - 2b_2 = 0 \quad (27)$$

The estimates of the green values at positions $x=1, 3$ and 5 are given by:

$$\hat{G}_1 = S_0(1) = a_0 + b_0 + c_0 + d_0 \quad (28)$$

$$\hat{G}_3 = S_1(3) = 27a_1 + 9b_1 + 3c_1 + d_1 \quad (29)$$

$$\hat{G}_5 = S_2(5) = 125a_2 + 25b_2 + 5c_2 + d_2 \quad (30)$$

As 12 equations are needed to obtain a unique solution, two extra conditions are required. Using the hue assumption [4] based on the correlation between colour planes, we propose the extra two equations be derived as follows:

By the hue assumption [4], we have

$$B_x - G_x = B_{x-2} - G_{x-2}. \text{ This implies}$$

$$B_3 - B_1 = \hat{G}_3 - \hat{G}_1 \quad (31)$$

Hence by (28) and (29)

$$B_3 - B_1 = 27a_1 + 9b_1 + 3c_1 + d_1 - a_0 - b_0 - c_0 - d_0 \quad (32)$$

$$\text{Similarly, } B_5 - B_3 = \hat{G}_5 - \hat{G}_3 \quad (33)$$

Hence by (29) and (30)

$$B_5 - B_3 = 125a_2 + 25b_2 + 5c_2 + d_2 - 27a_1 - 9b_1 - 3c_1 - d_1 \quad (34)$$

To solve the 12 simultaneous equations, namely (8)-(13), (18), (20), (25), (27), (32) and (34), the matrix representation (35) is used.

$$\mathbf{MC} = \mathbf{V} \quad (35)$$

where \mathbf{M} , \mathbf{C} and \mathbf{V} are given by (36).

$$\text{Therefore, } \mathbf{C} = \mathbf{M}^{-1}\mathbf{V} \quad (37)$$

where \mathbf{M}^{-1} is the inverse of the matrix \mathbf{M} . It can be readily shown that this inverse exists and can be pre-evaluated and stored. Hence this makes the algorithm computationally efficient. Finally the interpolated green value at the blue pixel position at $x=3$ can be evaluated by (29). Likewise, a missing green value at a red pixel position can be evaluated using similar equations. Similarly, an equation to determine the missing green value at the same position using vertical samples instead of horizontal samples can be derived.

$$\text{Let } \mathbf{M} = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 8 & 4 & 2 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 8 & 4 & 2 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 64 & 16 & 4 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 64 & 16 & 4 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 216 & 36 & 6 & 1 & 0 \\ 12 & 4 & 1 & 0 & -12 & -4 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 48 & 8 & 1 & 0 & -48 & -8 & -1 & 0 & 0 \\ 12 & 2 & 0 & 0 & -12 & -2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 24 & 2 & 0 & 0 & -24 & -2 & 0 & 0 & 0 \\ -1 & -1 & -1 & -1 & 27 & 9 & 3 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -27 & -9 & -3 & -1 & 125 & 25 & 5 & 1 & 0 \end{bmatrix}; \mathbf{C} = \begin{bmatrix} a_0 \\ b_0 \\ c_0 \\ d_0 \\ a_1 \\ b_1 \\ c_1 \\ d_1 \\ a_2 \\ b_2 \\ c_2 \\ d_2 \end{bmatrix}; \mathbf{V} = \begin{bmatrix} G_0 \\ G_2 \\ G_2 \\ G_4 \\ G_4 \\ G_6 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ B_3 - B_1 \\ B_5 - B_3 \end{bmatrix} \quad (36)$$

3. OUR PROPOSED ALGORITHM

To prevent interpolation across an edge in order to avoid blurring of an image, a selector based on the orientation of edges is used. An orientation map will first be produced from the CFA input [11]. For the green plane, cubic spline interpolation will then be applied to a set of vertical or horizontal samples accordingly. The flowchart in Fig. 3 shows our proposed algorithm for the green plane.

Since the red and blue colours are sampled at half the rate of the green colour, first order approximation is sufficient for the interpolation of the missing red and blue colour pixel values [10]. The flowcharts for the red and blue planes are similar to Fig. 3, except that first order interpolation [12] is used instead of cubic spline interpolation.

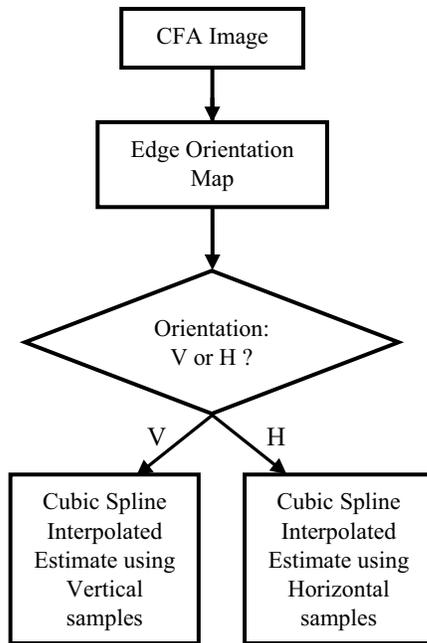


Fig. 3: Flowchart for Interpolation of Missing Green Values

4. EXPERIMENTAL RESULTS

Ten images with various characteristics, as shown in Fig. 4, were selected to evaluate our algorithm. Table 1 gives the quantitative measures of our proposed method when compared with other existing demosaicking methods [3,4,5,6,7,9,15]. It is evident that our method produced the smallest amount of error in the demosaicked output images. In order to evaluate our method visually, the picket-fence section of the well-known Lighthouse image was used. The picket fence section has been found to be a difficult area for CFA demosaicking as it has closely spaced vertical edges. Fig. 5 shows that our proposed method produced the least amount of colour artifacts visually, and Table 2 supports this visual result quantitatively.

Demosaicking Methods	NCD[8]	MSE
Bilinear	0.1165	0.1750
Freeman [3]	0.0656	0.1077
Kimmel [6]	0.0775	0.1193
Hamilton [5]	0.0300	0.0649
Plataniotis [9]	0.0720	0.1131
Lu&Tan [7]	0.0176	0.0344
Gunturk [4]	0.0166	0.0282
XLi [15]	0.0808	0.1136
Our Proposed Method	0.0103	0.0281

Table 2: Performance Measures of Demosaicking Methods

5. CONCLUSION

Our proposed method uses cubic spline interpolation in conjunction with a selector to avoid interpolating across an edge. This will prevent blurring of an edge. Two new equations based on the correlation between colour planes were derived to produce a unique solution for cubic spline interpolation. The inverse matrix for coefficient evaluation has been found to exist and can be pre-evaluated and stored for fast and efficient computation. From the experimental results, our proposed algorithm produced demosaicked images with minimal colour artifacts and least amount of errors when compared with other existing demosaicking techniques.

6. REFERENCES

- [1] B. E. Bayer. (1976): Color Imaging Array. US Patent 3 971 065.
- [2] D. R. Cok. (1987): Signal Processing Method and Apparatus for Producing Interpolated Chrominance Values in a Sampled Color Image Signal. US Patent 4 642 678.
- [3] W. T. Freeman. (1988): Median Filter for Reconstructing Missing Colour Samples. US Patent 4 724 395.
- [4] B. K. Gunturk, Y. Altunbasak, and R. M. Mersereau. (2002): Colour Plane Interpolation Using Alternation Projections. *IEEE Transactions on Image Processing*, 11:997-1013.
- [5] J. F. Hamilton Jr., and J. E. Adams Jr. (1997): Adaptive Colour Plan Interpolation in Single Sensor Colour Electronic Camera. US Patent 5 629 734.
- [6] R. Kimmel. (1999): Demosaicing: Image Reconstruction from Colour CCD Samples. *IEEE Transactions on Image Processing*, 8:1221-1228.
- [7] W. Lu and Y.-P. Tan. (2003): Colour Filter Array Demosaicking: New Method and Performance Measures. *IEEE Transactions on Image Processing*, 12:1194-1210.
- [8] K. N. Plataniotis and A. N. Venetsanopoulos. (2000): *Colour Image Processing and Applications*, Springer Verlag.
- [9] K. N. Plataniotis and R. Lukac. (2004): An Efficient Demosaicing Approach with a Global Control of Correction Steps. *IEEE International Conference on Acoustics, Speech, and Signal Processing Proceedings*, pp. III 469-472.
- [10] J. S. J. Li and S. Randhawa. (2005): High Order Extrapolation using Taylor Series for Color Filter Array Demosaicing. *Lecture Notes in Computer Science*, M. Kamel and A. Campilho, Eds., Springer-Verlag, 2005, 3656:703-711.

- [11] J. S. J. Li and S. Randhawa. (2005): Improved Accuracy for Colour Filter Array Demosaicking using High Order Extrapolation. Proceedings of ISSPA 2005, Sydney, 331-334.
- [12] S. Randhawa and J. S. J. Li. (2005): CFA Demosaicking with Improved Colour Edge Preservation. Proceedings of IEEE Tencon'05, 21-24 November 2005, Melbourne Australia.
- [13] J. S. J. Li and S. Randhawa. (2006): CFA Demosaicking by Adaptive Order of Approximation. Proceedings of VISAPP 2006, 25-28 February 2006, Setubal Portugal, 1:5-10.
- [14] J. S. J. Li and S. Randhawa. (2006): A Structural Approach to Improved Colour Filter Array Demosaicking for the Bayer Pattern, Proceedings of the 8th IASTED International Conference on Signal and Image Processing SIP 2006, 14-16 August 2006, Hawaii USA, 157-161.
- [15] X. Li. (2005): Demosaicking by Successive Approximation. IEEE Transactions on Image Processing, 14: 370-379.
- [16] C. De Boor. (1978): A Practical Guide to Splines. New York, Springer-Verlag.

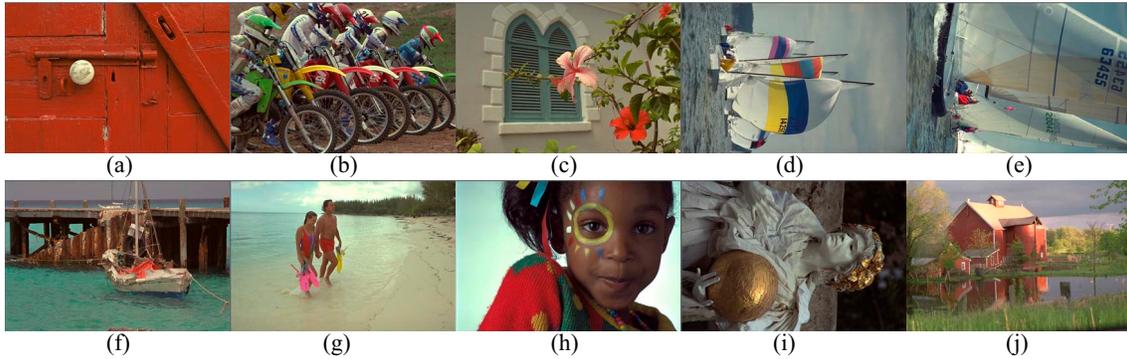


Fig. 4: Test images (a) – (j)

NCD[8]	Bilinear	Freeman [3]	Kimmel [6]	Hamilton [5]	Plataniotis [9]	Lu&Tan [7]	Gunturk [4]	XLi [15]	Our Proposed Method
Fig. 4(a)	5.47E-03	4.13E-03	5.83E-03	4.00E-03	3.93E-03	4.12E-03	4.74E-03	4.78E-03	3.41E-03
Fig. 4(b)	2.51E-02	8.76E-03	1.73E-02	1.23E-02	1.96E-02	7.74E-03	5.12E-03	1.65E-02	4.98E-03
Fig. 4(c)	7.69E-03	3.78E-03	4.66E-03	4.77E-03	6.04E-03	3.52E-03	3.57E-03	5.06E-03	2.55E-03
Fig. 4(d)	7.89E-03	3.53E-03	3.68E-03	3.41E-03	5.64E-03	2.42E-03	2.22E-03	4.25E-03	1.99E-03
Fig. 4(e)	6.95E-03	2.75E-03	3.16E-03	3.03E-03	4.57E-03	2.22E-03	2.33E-03	3.57E-03	2.02E-03
Fig. 4(f)	1.12E-02	6.62E-03	9.21E-03	7.07E-03	6.64E-03	5.54E-03	5.15E-03	8.68E-03	4.62E-03
Fig. 4(g)	3.56E-03	1.91E-03	2.28E-03	2.19E-03	2.60E-03	1.82E-03	1.80E-03	2.51E-03	1.66E-03
Fig. 4(h)	6.39E-03	3.53E-03	5.51E-03	3.59E-03	6.81E-03	3.10E-03	3.47E-03	6.26E-03	3.27E-03
Fig. 4(i)	9.92E-03	5.98E-03	7.27E-03	6.96E-03	8.60E-03	5.94E-03	5.39E-03	8.53E-03	4.63E-03
Fig. 4(j)	7.97E-03	4.62E-03	5.75E-03	4.77E-03	5.65E-03	4.19E-03	4.29E-03	5.56E-03	3.97E-03

Table 1: NCD Performance Measure for Test Images

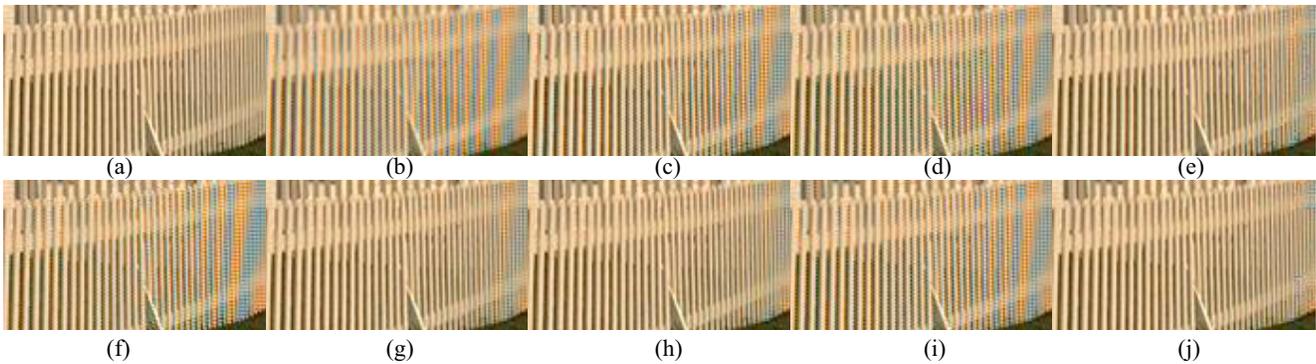


Fig. 5: (a) Picket fence region of the original Lighthouse image and the demosaicked output images using (b) Bilinear interpolation, (c) Freeman [3], (d) Kimmel [6], (e) Hamilton [5], (f) Plataniotis [9], (g) Lu&Tan [7], (h) Gunturk[4], (i) XLi [15] and (j) our proposed method.