RI-002

CFA Interpolation using Unified Geometry Map

伊藤 裕二

Yuji Itoh

1. INTRODUCTION

Single sensor digital cameras of consumer use generally employ color filter array (CFA) to represent multiple color spectral components, such as red, green, and blue. At the location of each pixel only one color sample is taken, and the other colors must be interpolated from neighboring samples. This color plane interpolation is known as CFA interpolation or demosaicking, and it is one of the important tasks in a digital camera pipeline. If CFA interpolation is not performed appropriately, consequent images would suffer from visible color artifacts.

The CFA interpolation methods can be grouped into two distinct categories. The first category is named “intra-color”, which applies interpolation techniques to each color channel separately. This category includes nearest-neighbor replication, bilinear interpolation, and cubic spline interpolation. It was pointed out that the intra-color algorithms often failed in handling non-stationary regions. Natural images, for which we are going to tune the CFA interpolation algorithm, show high cross-correlation between color channels as they represent reflection of source light against the same objects. Therefore, it is expected that better performance would be possible by exploiting correlation between color channels. The second category of algorithms called “inter-color” has been studied in this context. Therefore, we put more focus on inter-color methods in our research.

Having looking at various conventional CFA interpolation methods, we concluded that the bottom line must consist in how missing pixel is correlated with already available adjacent pixels (regardless of whether the algorithm is of intra-color or inter-color). The correlation is usually represented as weighting coefficients of the CFA interpolation filter that is mostly realized with weighted averaging. While the weighting coefficients are fixed in linear intra-color approaches, they are signal dependent in adaptive inter-color scheme, e.g. the weighting coefficients are inversely proportional to the local gradients in the edge based methods. Thus, we recognized that finding the optimal weighting coefficients is the most critical problem to derive an effective CFA interpolation. In this paper, we propose an innovative CFA interpolation scheme using so-called unified geometry map (UGM) intended to address the above problem.

2. CONVENTIONAL ALGORITHMS

First, we look into several major techniques in the inter-color CFA interpolation category. One approach in this category is smooth hue transition [1]. Smooth hue transition algorithms are based on the assumption that hue does not change abruptly between neighboring pixel locations. As a first step, these algorithms interpolate the green channel, which is usually done using bilinear interpolation. The red and blue samples are estimated from the bilinearly interpolated “red hue” (red-to-green ratio) and “blue hue” (blue-to-green ratio).

Another approach that exploits inter-color correlation is edge-directed interpolation [2], [3]. The main difference between this approach and the previous one is that the bilinear interpolation of the green channel is replaced by adaptive interpolation to prevent interpolating across edges. In [2], first-order horizontal and vertical gradients are computed at each missing green location on the Bayer pattern. Instead of interpolating color differences or color ratios, it is also possible to use the inter-color correlation as a correction term in the interpolation. In [3], Hamilton and Adams used second-order derivatives of the chrominance samples as correction terms in the green channel interpolation.

There are also more complicated demosaicking approaches. In [4], Gunturk et. al. proposed a unique method called alternating projection (AP) that utilizes inter-channel correlation. Their algorithm defines constraint sets based on the observed color samples and prior knowledge about the correlation between the channels, and reconstructs the color channels by projecting the initial estimates onto these constraint sets.

In a recent literature [5], Lukac et. al. proposed so-called data adaptive filters (DAF) and compared its performance with the prior arts described above. It was revealed in [5] that the DAF outperformed the other algorithms except the AP that is almost comparable to the DAF schemes. Since the AP is far more complicated than the DAF, we chose the DAF as target. It was found that the uniqueness of the DAF algorithm consisted in: a) spectral modeling and b) definition of distance between samples in the same color plane. As for the spectral modeling, four major models: 1) color-difference model (CDM), 2) color-ratio model (CRM), 3) normalized CRM utilizing the linear shifting operations (NRSM), and 4) normalized CRM utilizing both linear scaling and shifting operations (NRSSM), were proposed so far. Among these spectral models in DAF, the color difference model (CDM) performed the best according to [5]. Thus, we chose the DAF-CDM as target. Let $\oplus$ be a relation between the two inputs A and B in terms of the CDM and CRM, and $\otimes$ be an inverse operator of $\oplus$. These operators are defined as:

$$\begin{aligned} A\oplus B &= A-B \text{ and } A\otimes B = A+B \quad \text{for CDM} \\ A\oplus B &= A/B \text{ and } A\otimes B = A\cdot B \quad \text{for CRM} \end{aligned} \tag{1}$$

Here let us consider the population of the RGB vectors $\boldsymbol{x}_{(i,j)} = [x_{(i,j)0}, x_{(i,j)1}, x_{(i,j)2}]$ with $x_{(i,j)k}$ indicating the R (k=0), G (k=1) and B (k=2) component at coordinates (i, j). The distance at coordinates (i, j) in k-th color plane, which is denoted by $d_{(i,j)k}$, is obtained as follows.

$$d_{(i,j)k} = \sum_{(g,h)\in\zeta} \left| x_{(i,j)k} - x_{(g,h)k} \right| \tag{2}$$

, where ζ represents the local vicinity around the coordinates (i, j)

Now let $\sigma_{(i,j)k}$ denote similarity measure at coordinates (i, j) in k-th color plane, which is generally expressed as a function of $d_{(i,j)k}$. Among several candidates, Lukac derived a rather simple realization that led to the best performance in terms of interpolation fidelity, that is:

$$\sigma_{(i,j)k} = \frac{1}{1+ d_{(i,j)k}} \tag{3}$$

Finally, the weighting coefficients of adaptive interpolation filter to be used later, which is denoted by $w_{(i,j)k}$, is given by the normalization formula:

$$w_{(i,j)k} = \frac{\sigma_{(i,j)k}}{\sum_{(g,h)\in\zeta} \sigma_{(g,h)k}} \tag{4}$$

日本テキサス・インスツルメンツ株式会社

Now let us define the data adaptive filtering (DAF), which still belongs to component-wise processing by itself (i.e. intra color processing), using Equation 4 as follows.

$$x_{(s,t)k} = \sum_{(i,j)\in\varsigma} w_{(i,j)k} \cdot x_{(i,j)k} \tag{5}$$

Eventually, the DAF-CDM is derived by combining the generalized spectral model and the data adaptive filter set forth above:

$$x_{(s,t)k} = x_{(s,t)1} \otimes \sum_{(i,j)\in\varsigma} w_{(i,j)k} \cdot \left(x_{(i,j)k} \oplus x_{(i,j)1}\right) \quad for\ k = 0\ and\ 2 \tag{6}$$

$$x_{(s,t)1} = x_{(s,t)k} \otimes \sum_{(i,j)\in\varsigma} w_{(i,j)k} \cdot \left(x_{(i,j)1} \oplus x_{(i,j)k}\right) \quad for\ k = 1 \tag{7}$$

3. PROPOSED ALGORITHM

3.1 Concept of proposed algorithm

As we saw in the previous sections, the DAF-CDM approach has two main features: 1) spectral modeling (i.e. CDM) and 2) edge-sensing algorithm (i.e. DAF). Among the two main features above, the proponents seemed to put a primary focus on the edge-sensing mechanism. Then, we looked into the details of the edge-sensing mechanism that the DAF-CDM adopted, and identified two issues as described below.

One issue is that the DAF edge-sensing method doesn't exploit inter-color correlation. As expressed in Equations 2, 3, 4 and 5, the weighting coefficients to be applied in the interpolation filter are derived in intra-color fashion. This led us to an idea that tailoring inter-color model would probably improve the performance of CFA interpolation. Incidentally, the CDM and CRM were derived based on the phenomenon that signals at one color plane are highly correlated with those of the other color planes. This implies a possibility of creating metadata that represents gray values common to all color planes. In this context, we introduced so-called unified geometry map (UGM). The UGM is 1-bit plane, where value at each pixel, called map index, indicates whether the pixel belongs to relatively dark one or bright. Let $m_{(i,j)}$ denote map index at coordinates (i, j). Once the map index is obtained somehow, it is used in CFA interpolation as below. Fig. 1 illustrates UGM based CFA interpolation, where the pixel to be interpolated is surrounded by three bright pixels and one dark pixel. The corresponding map indices shown abreast indicate that the pixel to be interpolated belongs to the same category as the dark pixel. The UGM based scheme applies larger weight to the dark pixel and smaller weights to the three bright pixels, which is a genuine inter-color CFA interpolation. On the other hand, in case of the DAF-CDM larger weighting coefficient is applied to three bright pixels rather than the dark pixel as opposed to the UGM based scheme. This is because the weighting coefficients are calculated based on a sort of majority rule in the DAF algorithm, that is, dark pixel has less contribution to the filter output when bright pixels are dominant in this case.

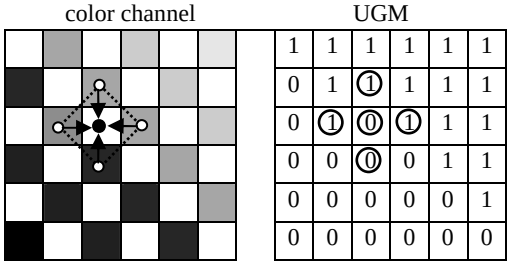


Fig. 1: UGM based CFA interpolation.

The other issue is that the DAF edge-sensing scheme requires a considerable amount of computational resources. These computations are spent on calculating the weighting coefficients based on the Equations 2, 3 and 4. For implementation ease, we calculate the UGM in two-dimensional block with *M* by *N* pixels (horizontally *M* pixels and vertically *N* pixels), which is called UGM window. However, the larger the UGM window is the more computations are required. We came to a pertinent solution against the computation issue, i.e. sharing UGM while applying CFA interpolation to all missing pixels in the UGM window. Thus, we can significantly reduce the computational complexity, which is quantitatively expressed later.

Having derived likely solutions against the two issues above, we had to tailor a spectral model or choose one among several candidates (e.g. CDM, CRM). We chose the CDM in the sense that we can do a fair comparison with the target in terms of the edge-sensing algorithm. Another reason to adopt the CDM is that it requires much less resources than the CRM while these two bring almost similar performance. Fig. 2 shows the block diagram of the proposed algorithm set forth above.

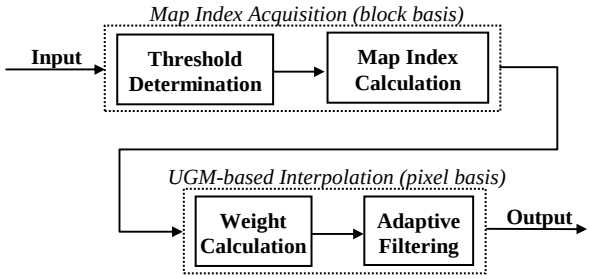


Fig. 2: Block diagram of pro posed algorithm.

3.2 Embodiment of proposed algorithm

Here we embody the concept of the proposed algorithm set forth in the previous section. The algorithm is basically two-fold: 1) map index acquisition and 2) UGM based interpolation. In this embodiment, we assume that the input data are compatible with Bayer CFA pattern.

3.2.1 Map index acquisition

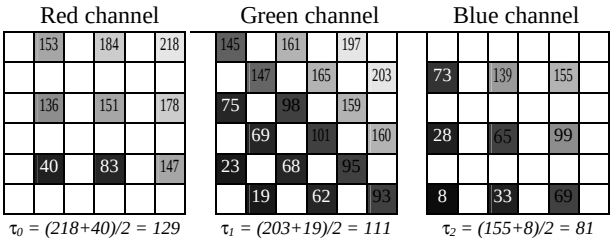
The map indices are obtained on a UGM window basis with a threshold specific to the input CFA data in the *M* by N block. In each UGM window, a threshold value shall be determined first. In an illustration below, we employ middle of the signal dynamic range in 6x6 block (*M*=*N*=6) as threshold. Note that the threshold is set per each color component in a block. Let max_k and min_k be the maximum and minimum values in a block for color *k* (*R* for *k*=0, *G* for *k*=1, *B* for *k*=2), respectively. So we define the threshold denoted by τ_k:

$$\tau_k = \left(\max_k + \min_k\right)/2 \tag{8}$$

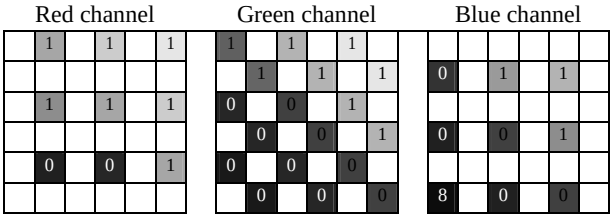
Now let $\lambda_{(i,j)}$ denote map index at coordinates (i, j). The map index $\lambda_{(i,j)}$ is determined based on whether the pixel value $x_{(i,j)k}$ is greater than the threshold or not, which is given below.

$$\lambda_{(i,j)} = \begin{cases} 1 & if\ x_{(i,j)k} \geq \tau_k \\ 0 & otherwise \end{cases} \tag{9}$$

As in Equation 9 above, the map index is not dependent on color component. In addition, the map indices provide a very rough idea of object surface. That's why this was named unified geometry map. An illustrative processing flow of the map index acquisition is depicted in Fig. 3.



(a) Calculation of threshold: τ_k (input gray level appears inside pixel)



(b) Calculation of map index: $\lambda_{(i,j)}$

UGM

1	1	1	1	1	1
0	1	1	1	1	1
0	1	0	1	1	1
0	0	0	0	1	1
0	0	0	0	0	1
0	0	0	0	0	0

(c) Integration of three map index planes into one UGM

Fig. 3: Illustrative processing flow of map index acquisition.

3.2.2 UGM based interpolation

Once the map indices are obtained, an interpolation filter is applied to all relevant pixels in the UGM window (i.e. *M by N* pixel block). The tasks are two-fold: 1) calculation of weighting coefficients, and 2) adaptive filtering. As we designed in the concept, larger weights are applied to the neighboring pixels that have the same map index compared with those that have the opposite map index. Thus, the weighting coefficients are calculated accordingly. An example set of the weighting coefficients is given in Table 1.

Table 1: Example set of the weighting coefficients.

Mode	condition of map indices *1				weighting coefficients: $w_{(i,j)k}$ *2			
	λ_a *XOR* λ_b	λ_a *XOR* λ_c	λ_a *XOR* λ_d	λ_a *XOR* λ_e	b	c	d	e
0	0	0	0	0	4	4	4	4
1	0	0	0	1	5	5	5	1
2	0	0	1	0	5	5	1	5
3	0	0	1	1	6	6	2	2
4	0	1	0	0	5	1	5	5
5	0	1	0	1	6	2	6	2
6	0	1	1	0	6	2	2	6
7	0	1	1	1	10	2	2	2
8	1	0	0	0	1	5	5	5
9	1	0	0	1	2	6	6	2
10	1	0	1	0	2	6	2	6
11	1	0	1	1	2	10	2	2
12	1	1	0	0	2	2	6	6
13	1	1	0	1	2	2	10	2
14	1	1	1	0	2	2	2	10
15	1	1	1	1	4	4	4	4
Note	*1: *XOR* represents exclusive OR operation *2: The weights sum up to 16, so the result has to be right shifted by 4 bits							

Note that Table 1 is primarily designed for 4 pixel input case (ζ = {(s-1, t-1), (s-1, t+1), (s+1, t-1), (s+1, t+1)} or ζ = {(s-1, t), (s, t-1), (s, t+1), (s+1, t)}), where λ_a and {λ_b, λ_c, λ_d, λ_e} denote the map index of the pixel to be interpolated (i.e. a) and the map indices of the input pixels, respectively. Fig. 4 shows illustrative explanation of the UGM based interpolation process after the map indices are obtained (i.e. A-stage). The processing at each stage is described in the following sections.

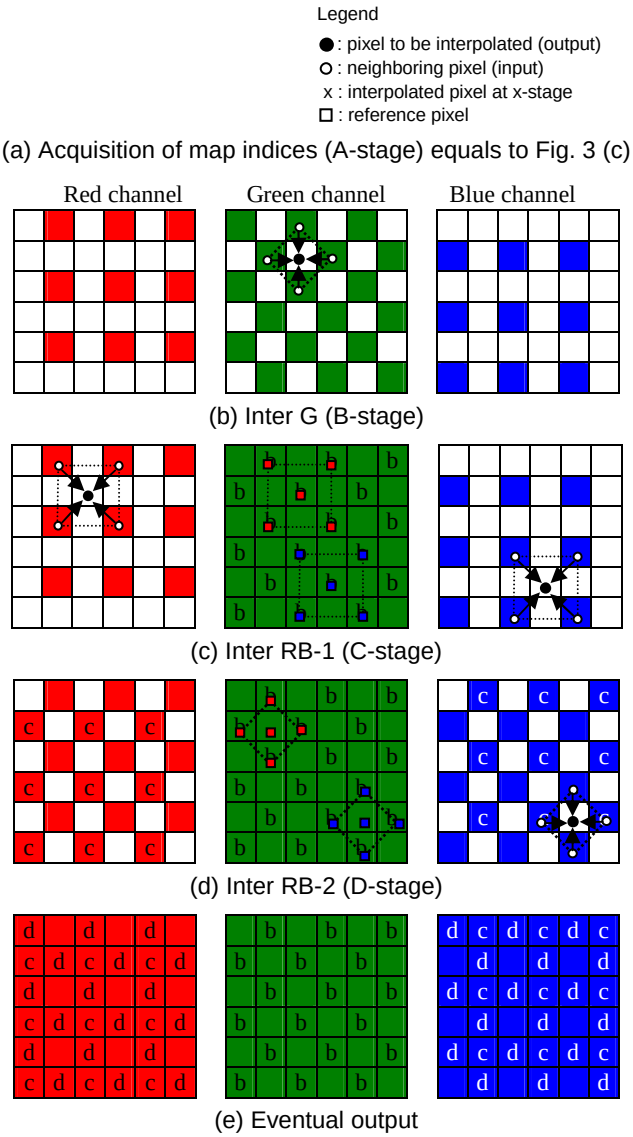


Fig. 4: Processing flow of UGM based interpolation.

3.2.2.1 Inter G (B-stage)

Let ζ = {(s-1, t), (s, t-1), (s, t+1), (s+1, t)} in Fig. 4 (b). In the DAF-CDM method, the first G component interpolation is done in intra-color fashion. The proposed method employs inter-color interpolation based on the UGM, which provides better quality than intra-color scheme in most cases. This is one of the major advantages of the UGM method over conventional algorithms. First, calculate the weighting coefficients $w_{(i,j)k}$ with k=1 according to Table 1 and the map indices. The missing G component at coordinates (s, t) in this stage is interpolated according to Equation 7. The entire G plane is filled up after this stage is applied to all relevant G pixels. Thereafter, the map

indices are updated based on G signals as shown in Section 3.2.1. Note that R and B components are left untouched in this stage.

3.2.2.2 Inter RB-1 (C-stage)

Let ζ = {(s-1, t-1), (s-1, t+1), (s+1, t-1), (s+1, t+1)} in Fig. 4 (c). Calculate the weighting coefficients $w_{(i,j)k}$ with k=0 according to Table 1 and the map indices. The missing R component at coordinates (s, t) in this stage is interpolated in inter color fashion according to Equation 6. Similarly, interpolate the missing B components with k=2. The half population of both R and B planes are filled up after this stage is applied to all relevant pixels. Note that G components are left untouched in this stage.

3.2.2.3 Inter RB-2 (D-stage)

Let ζ = {(s-1, t), (s, t-1), (s, t+1), (s+1, t)} in Fig. 4 (d). Calculate the weighting coefficients $w_{(i,j)k}$ with k=0 according to Table 1 and the map indices. The missing R component at coordinates (s, t) in this stage is interpolated in inter color fashion according to Equation 6. Similarly, interpolate the missing B components with k=2. The whole population of both R and B planes are filled up after this stage is applied to all relevant pixels. The G components are left untouched in this stage, but they are merely referred in the inter color processing.

3.3 Complexity analysis

Table 2 summarizes the normalized computations required to interpolate one missing pixel with regard to basic arithmetic and logic operations, in which we compare the proposed method with DAF-CDM and DAF-CRM. Apparently, the proposed algorithm requires much less computation compared with the DAF-CDM and DAF-CRM.

Table 2: Normalized computations per pixel interpolation.

operation / algorithm	shift	add / sub	cmp	abs	mpy	div
DAF-CDM	0	29	0	6	4	8
DAF-CRM	0	24	0	6	5	12
Proposed (M=N=8 case)	5.1	7.1	6.4	0	3	0

4. EXPERIMENTS

In the experiments, we measure the performance of the proposed method, DAF-CDM and DAF-CRM, and compare them quantitatively. We use three objective quality criteria: 1) the mean absolute error, 2) the mean square error (MSE), and 3) the normalized color difference (NCD), which are defined in Equation 10. The NCD criterion was intended to measure the difference between the original image and processed image in the units corresponding to human perception.

$$MAE = \frac{1}{3ST}\sum_{k=0}^{2}\sum_{s=0}^{S-1}\sum_{t=0}^{T-1}\left|o_{(s,t)k} - x_{(s,t)k}\right| \quad (10)$$

$$MSE = \frac{1}{3ST}\sum_{k=0}^{2}\sum_{s=0}^{S-1}\sum_{t=0}^{T-1}\left(o_{(s,t)k} - x_{(s,t)k}\right)^2$$

$$NCD = \frac{\sum_{s=0}^{S-1}\sum_{t=0}^{T-1}\sqrt{\sum_{k=0}^{2}\left(o'_{(s,t)k} - x'_{(s,t)k}\right)^2}}{\sum_{s=0}^{S-1}\sum_{t=0}^{T-1}\sqrt{\sum_{k=0}^{2}\left(o'_{(s,t)k}\right)^2}}$$

, where $\boldsymbol{o}_{(s,t)} = [o_{(s,t)0}, o_{(s,t)1}, o_{(s,t)2}]$ and $\boldsymbol{x}_{(s,t)} = [x_{(s,t)0}, x_{(s,t)1}, x_{(s,t)2}]$ denote the RGB vectors at coordinated (s,t) of original image and restored image (both images are of extension SxT), respectively. Also $\boldsymbol{o'}_{(s,t)} = [o'_{(s,t)0}, o'_{(s,t)1}, o'_{(s,t)2}]$ and $\boldsymbol{x'}_{(s,t)} = [x'_{(s,t)0}, x'_{(s,t)1}, x'_{(s,t)2}]$ represent the RGB vectors $\boldsymbol{o}_{(s,t)}$ and $\boldsymbol{x}_{(s,t)}$ in the CIE LUV color space with the white point D65.

The test images used are shown in Fig. 5. Fig 5 (a) was captured by a 3CCD camera, and Fig. 5 (b) is a standard image called “Baboon” that contains a lot of color transitions (hence, adequate for CFA interpolation test).

(a) Geese (320 by 240 pixels)

(b) Baboon (512 by 512 pixels)

Fig. 5: Test images used for experiments.

Table 3 summarizes the simulation results, which revealed that the proposed method outperformed DAF-CDM and DAF-CRM at any column (i.e. any image and any criterion). It was also observed that the DAF-CDM is superior or comparable to the DAF-CRM. This justified the use of the DAF-CDM as target.

Table 3: Simulation results.

image	Geese			Baboon		
criteria / algorithm	MAE	MSE	NCD	MAE	MSE	NCD
DAF-CDM	5.15	98.9	0.0662	8.96	246.7	0.125
DAF-CRM	5.34	105.2	0.0678	9.15	256.0	0.124
Proposed (M=N=8 case)	4.83	83.2	0.0641	8.39	215.0	0.118

5. CONCLUSIONS

This paper has described an innovative CFA interpolation technique that uses unified geometry map, which is intended to produce high fidelity picture with relatively small amount of resource consumption. We confirmed through the simulations that the proposed algorithm outperformed the state-of-the-arts technologies, namely DAF-CDM and DAF-CRM, in terms of objective quality measures. In addition, our scheme spends less resource than the state-of-the-arts technologies.

References

[1] 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, Feb. 1987.

[2] R. H. Hibbard, “Apparatus and method for adaptively interpolating a full color image utilizing luminance gradients,” U.S. Patent 5 382 976, Jan.1995.

[3] J. F. Hamilton Jr. and J. E. Adams, “Adaptive color plane interpolation in single sensor color electronic camera,” U.S. Patent 5 629 734, May1997.

[4] B. Gunturk, Y. Altunbasak, and R. Mersereau, “Color plane interpolation using alternating projections,” IEEE Transaction on Image Processing, vol. 11, no. 9, pp. 997-1013, Sept. 2002.

[5] R. Lukac and K. N. Plataniotis, “Data adaptive filters for demosaicking: A framework,” IEEE Transaction on Consumer Electroics, vol. 51, no. 2, pp. 560-570, May. 2005.