A Practical CFA Interpolation Using Local Map

Yuji ITOH*), Member

SUMMARY This paper introduces a practical color filter array (CFA) interpolation technique. Among the many technologies proposed in this field, the inter-color methods that exploit correlation between color planes generally outperform the intra-color approaches. We have found that the filtering direction, e.g., horizontal or vertical, is among the most decisive factors for the performance of the CFA interpolation. However, most of the state-of-the-art technologies are not flexible enough in determining the filtering direction. For example, filtering only in the upper direction is not usually supported. In this context, we propose an inter-color CFA interpolation using a local map called unified geometry map (UGM). In this method, the filtering direction is determined based on the similarity of the local map data. Thus, it provides more choices of the filtering directions, which enhances the probability of finding the most appropriate direction. It is confirmed through simulations that the proposal outperforms the state-of-the-art algorithms in terms of objective quality measures. In addition, the proposed scheme is as inexpensive as the conventional methods with regard to resource consumption.

**key words:** color filter array interpolation, inter-color correlation, color difference model, unified geometry map

1. Introduction

Single sensor digital cameras for 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, which is one of the important tasks in a digital camera pipeline. Since ill-designed CFA interpolation would suffer from visible color artifacts, a lot of studies have been made in this field.

The most commonly used color pattern is Bayer pattern [1]. As seen in Fig. 1, in a Bayer pattern, green samples are obtained on a quincunx lattice (checkerboard pattern), and red and blue samples are on rectangular lattices. The density of the red and blue samples is one-half that of the green ones.

In the last two decades, various CFA interpolation methods have been proposed. These methods can be classified into two groups: iterative and non-iterative. Although the iterative algorithms [2]–[4] generally outperform the non-iterative ones, they are not suited for consumer products as a practical solution not only because they tend to be resource hungry but also because the resource requirements cannot be fixed at design time. Therefore, we don’t handle the iterative algorithms in this paper.

There is another grouping. The CFA interpolation methods, both iterative and non-iterative, 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, bi-linear, bi-cubic, and B-spline. The interpolation filtering varies from algorithm to algorithm, and it is usually straightforward. Although the intra-color algorithms can provide satisfactory results in smooth regions of an image, they tend to fail in non-stationary regions. In the meantime natural images, for which we are going to tune the CFA interpolation algorithm, show high cross-correlation between color channels as they represent the reflection of the same objects. Therefore, it is expected that better performance would be attained by exploiting correlation between color channels. The second category of algorithms called “inter-color” has been studied in this context, and it has actually provided significantly better performance than the first category. Therefore, we put more focus on inter-color methods in this paper.

We investigated various conventional CFA interpolation methods, and found that finding the optimal direction of the interpolation filtering is the most critical problem to derive an effective CFA interpolation. The conventional inter-color methods [5]–[11] apply the interpolation filtering either in horizontal-, vertical-, diagonal- or omni-direction (i.e., no preferred orientation) based on the decision of some orientation classifier. These methods are not optimal in the sense that the filtering cannot be applied in other direction than the pre-determined choices of the filtering directions.

In this paper, we propose a practical inter-color CFA interpolation scheme that uses a local map called unified geometry map (UGM) [12]. The UGM is an inter-color local

---

Fig. 1 Bayer CFA pattern with GRGR first row.
map that is obtained from CFA data with a relatively simple operation. The proposed algorithm utilizes the UGM to decide which pixels are more correlated with the missing pixel. Thus, we have realized an inter-color CFA interpolation technique with flexible filtering directions. The rest of the paper is organized as follows. Section 2 reviews prior studies in this field. Section 3 describes the concept and embodiment of the proposed algorithm, which is evaluated in Sect. 4. Finally we make some concluding remarks in Sect. 5.

2. Conventional Inter-Color CFA Interpolation

Here in this section, we look into several major techniques in the inter-color CFA interpolation category, which will be evaluated in the experiments later. One approach in this category is the edge-directed interpolation [7]–[11]. The main feature of this approach consists in the edge-directed adaptive interpolation intended to prevent interpolating across edges. In [9], Hamilton and Adams used second-order derivatives of the chrominance samples as correction terms in the green channel interpolation. To determine the gradient at a red (or blue) sample location, the second-order derivative of red (or blue) pixel values is added to the first-order derivative of the green pixel values. The second-order derivative of the red (or blue) pixels is also added to the average of the green pixel values in the minimum gradient direction. The methods in [10], [11] are also based on the inter-color correlation, more precisely the color difference correlation. Kuno et al. claim that their methods are designed to function even in case of negative or no correlation (as well as positive correlation) whereas the conventional methods work only in case of high positive correlation. The conventional inter-color methods mentioned above apply the interpolation filtering either in horizontal-, vertical-, diagonal- or omni-direction (i.e., no preferred orientation) based on the decision of some orientation classifier. Here typical orientation classifiers are mathematically explained with the CFA data depicted in Fig. 1. The typical orientation classifiers at the R23 position representing signal discontinuity in horizontal direction, denoted by \( DH \), are Eq. (1) for [9] and Eq. (2) for [10], [11].

\[
DH = |2 \cdot R_{23} - R_{21} - R_{25}| + |G_{22} - G_{24}|
\]

\[
DH = \left[\frac{|2 \cdot R_{23} - R_{21} - R_{25}|}{2} + |G_{22} - G_{24}| + |2 \cdot G_{13} - G_{11} - G_{15}|/4 + |2 \cdot G_{33} - G_{31} - G_{35}|/4 + |(B_{12} - B_{14}) + (B_{32} - B_{34})|/2\right]/4
\]

The above orientation classifiers are intended to sense the high spatial frequency information present in the pixel neighborhood in the horizontal direction. The orientation classifiers in the other directions, such as one for the vertical direction denoted by \( DV \), are calculated similarly in respective direction. The filtering direction is, then, regulated by using the orientation classifiers according to Eq. (3).

\[
\begin{align*}
\text{If } DH &< DV \\
& \text{filtering horizontally} \\
\text{else if } DV &< DH \\
& \text{filtering vertically} \\
\text{else} \\
& \text{filtering omni–directionally}
\end{align*}
\]

Here we argue that these methods are not optimal in the sense that the filtering direction is chosen from among the pre-determined candidates even when the highest correlation exists in other direction that is not covered by the candidates.

Lukac et al. proposed so-called data adaptive filters (DAF) [13]. The uniqueness of the method in [13] mainly comes from an edge-sensing classifier called aggregated absolute differences, based on which the filtering masks (i.e., positions of the filter input pixels) and weighting coefficients are determined in a signal adaptive manner. This unique concept allows the filtering masks to be more flexible whereas most of the conventional methods [7]–[11] select one among several pre-determined masks that are represented as filtering direction such as horizontal or vertical.

Figure 2 shows an interpolation processing, which is common to all the inter-color methods. This corresponds to one stage (The entire algorithm is usually performed with multiple sequential stages), and illustrates which pixels are involved in the interpolation: i) missing pixel (black circle), ii) intra-color input pixels (white circle), and iii) inter-color input pixels (white rectangular). Here let us consider the population of the RGB vectors \( x(i, j) = [x(i, j)_R, x(i, j)_G, x(i, j)_B] \) with \( x(i, j)_k \) indicating the \( R \) (\( k = 0 \)), \( G \) (\( k = 1 \)) and \( B \) (\( k = 2 \)) component at coordinates \( (i, j) \). The aggregated absolute differences at coordinates \( (i, j) \) in \( k \)-th color plane, which is denoted by \( d(i, j)_k \), is obtained as follows.

\[
d(i, j)_k = \sum_{(g, h) \in \zeta} |x(i, j)_k - x(g, h)_k|
\]

where \( \zeta \) represents the local vicinity, which is equivalent to the intra-color input pixels, around the missing pixel. Then, the weighting coefficients denoted by \( w(i, j)_k \) of the adaptive interpolation filter is given based on the aggregated absolute
is performed by Eq. (6) for \( R \) and \( B \) components and \( G \) component, respectively. (Note \( k = 0 \) or 2):

\[
x(s, t)_k = x(s, t)_1 + \sum_{(i,j) \in S} w(i, j)_k \cdot (x(i, j)_k - x(i, j)_1) : R \text{ and } B
\]

\[
x(s, t)_1 = x(s, t)_k + \sum_{(i,j) \in S} w(i, j)_1 \cdot (x(i, j)_1 - x(i, j)_k) : G \quad (6)
\]

The DAF-CDM is apparently an inter-color method as shown by Eq. (6). However, the weighting coefficients are calculated in an intra-color fashion by Eqs. (4) and (5). Since the weighting coefficients are applied to the signal difference between the color channels, we suppose that they should be obtained in an inter-color manner. In short, the DAF-CDM doesn’t take full advantage of the inter-color correlation although it provides a sophisticated framework with the signal adaptive filtering mechanism.

3. Proposed CFA Interpolation Algorithm

This section details how the proposed algorithm realizes a practical inter-color CFA interpolation with a flexible filtering mechanism. As we reviewed in the previous section, the DAF edge-sensing method doesn’t fully exploit inter-color correlation. As expressed in Eq. (4) through Eq. (6), the weighting coefficients to be applied in the interpolation filtering are calculated in an intra-color fashion. In the meantime, the CDM spectral model has been 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 introduce a local map called unified geometry map (UGM) [12]. The UGM is 1-bit plane metadata (it can be extended to multi-bit plane), where value at each pixel, called map index, indicates whether the pixel belongs to relatively dark or light region. Once the map indices are obtained, they are used in the interpolation process.

Figure 3 illustrates the UGM based CFA interpolation, where the missing pixel is surrounded by three light 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 data suggest that the dark pixel is more correlated with the missing pixel. Thus, we will be able to achieve an inter-color CFA interpolation. On the other hand, in case of the DAF-CDM larger weighting coefficient is applied to the three light pixels (rather than the dark pixel) as opposed to the UGM based scheme. This is attributed to the fact that the DAF algorithm calculates the weighting coefficients based on a sort of majority rule, that is, dark pixel has less contribution to the filter output when light pixels are dominant.

In the meantime, the resource requirements are also critical for practical solutions. The adaptive methods such as the DAF require a considerable amount of computational resources. The interpolation filtering itself is expensive. However, a significant fraction of the resources are often spent on calculating the edge orientation classifiers [7]–[11] and the weighting coefficients. We call them “discriminators”, hereafter. Ideally these discriminators should represent the similarities between the missing pixel and its adjacent pixels. The UGM is metadata that convey a rough idea of the similarities between the pixels. It would save a great amount of computations if we are able to easily calculate the UGM and derive similarity measure based on the UGM. We found through preliminary experiments that the size of the region in which the UGM data is obtained shall be sufficiently larger (probably 2x or more) than the size of the interpolation filter to be applied later. If we look at only a small area in obtaining such metadata, it will lead us to so-called local minima/maxima problem [14]. 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 resource problem, that is, sharing UGM while applying CFA interpolation to all the missing pixels in the UGM window. In this way, we can significantly reduce the computational complexity, which is quantitatively expressed later (Table 2).

Figure 4 shows the block diagram of the proposed algorithm set forth above. The algorithm is basically three-fold: i) map index acquisition, ii) global edge classification, and iii) UGM based interpolation. Here we assume that the input data are compatible with Bayer CFA pattern shown in Fig. 1. Note that we discuss the global edge classification after the map index acquisition and the UGM based interpolation for explanation ease.

An illustrative processing flow of the map index acquisition is depicted in Fig. 5. 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,
Fig. 4  Block diagram of proposed algorithm.

![Block diagram of proposed algorithm.](image)

The threshold value shall be determined first. As shown by the illustration in Fig. 5, we employ the average gray scale of pixels within a UGM window (M = N = 6 in Fig. 5) as threshold. Note that the threshold is set per color component. Let x(i, j)k be the original pixel at (i, j) in a UGM window for color k (R for k = 0, G for k = 1, B for k = 2, k depends on the sample position). Here, we calculate the threshold denoted by τk:

\[ \tau_k = \frac{1}{F_k} \sum_{(i, j)} x(i, j)_k \]  

(7)

\( F_k \): the number of k colo pixels within a UGM window

Now let λ(i, j) denote map index at coordinates (i, j). The map index λ(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.

\[ \lambda(i, j) = \begin{cases} 1 & \text{if } x(i, j)_k \geq \tau_k \\ 0 & \text{otherwise} \end{cases} \]  

(8)

The map index is calculated based on the color-dependent threshold as indicated in Eq. (8). However, once the map indices are obtained, they are used in a color-independent fashion. In fact, the map indices provide a very rough idea of object surface, which is not dependent on color. That’s why this is named unified geometry map.

After the map indices are calculated, an interpolation filter is applied to all the relevant pixels in the UGM window (i.e., M by N pixel block). The tasks are two-fold: i) calculation of similarity, and ii) adaptive filtering. Now let’s consider what information the UGM tells us. It has been found that the pixels with the same map index (0 or 1) resemble in gray level even across the color channels as shown in Fig. 5. Then, we are going to exploit this useful feature.

An example of a missing G pixel interpolation is explained with the arrangement of CFA data and UGM depicted in Fig. 6. Gc is a missing pixel of the G channel located with Ac, where A is either R or B component. In the following equations, θ represents the similarity measure, and ζ denotes the local vicinity, which is equivalent to the input pixels, around the missing pixel (Fig. 7 depicts ζ in respective case).

Now we calculate Gc by:

\[ G_c = \sum_{l=0}^{3} \left[ \theta_l \cdot \left( G_l + (A_c - A_l)/2 \right) + \bar{\theta}_l \cdot \mu \right] \]  

(9)

\[ \theta_l = \begin{cases} 1 & \text{if } \lambda_l = \lambda_c \\ 0 & \text{otherwise} \end{cases} \]  

\[ \mu = \frac{1}{4} \sum_{l=0}^{3} [G_l + (A_c - A_l)/2] \]

The missing R (k = 0) and B (k = 2) pixels are obtained by:

\[ x(s, t)_k = \frac{x(s, t)_1 + \left[ \theta(i, j) \cdot (x(i, j)_k - x(i, j)_1) + \bar{\theta}(i, j) \cdot \rho \right]}{4\rho + \sum_{(i, j) \in \zeta} \theta(i, j) \cdot (x(i, j)_k - x(i, j)_1) + \bar{\theta}(i, j) \cdot \rho} \]  

(8)

\[ \theta(i, j) = \begin{cases} 1 & \text{if } \lambda(i, j) = \lambda(s, t) \\ 0 & \text{otherwise} \end{cases} \]
(i.e., along the edge) in most cases. However, it occasionally misleads to filtering horizontally (i.e., across the edge), which degrades the performance. Therefore, we introduce a measure based on a classifier called "global edge classifiers" that sense the structure of a broader region in order to avoid the above failure. The global edge classifiers, denoted by $DH_{ge}$ and $DV_{ge}$, are calculated on the UGM window by:

$$DH_{ge} = \sum_{i=0}^{N-1} \sum_{j=0}^{M-2} f(\lambda(i, j), \lambda(i, j + 1))$$

$$DV_{ge} = \sum_{i=0}^{N-1} \sum_{j=0}^{M-2} f(\lambda(i, j), \lambda(i + 1, j))$$

The similarity measure is then, regulated by using the global edge classifiers according to Eq. (12). This process, i.e., enforcing particular similarity measures to be zero, is applied to Eq. (9) in calculating all the missing $G$ pixels in the block concerned.

If $DH_{ge} + \delta < DV_{ge}$

$$\theta_0 = \theta_3 = 0$$

else if $DV_{ge} + \delta < DH_{ge}$

$$\theta_1 = \theta_2 = 0$$

Figure 7 shows the UGM based interpolation process after the map index acquisition and the global edge classification are performed. Note that the acquisition of the map indices (A-stage), which equals to Fig. 5, is omitted in Fig. 7.

3.1 Inter G (B-Stage)

In Fig. 7 (b), let $\{G_c, G_0, G_1, G_2, G_3\} = \{x(s, t), x(s - 1, t), x(s, t - 1), x(s, t + 1), x(s + 1, t), x(s + 1, t + 1)\}$, and $\{X_s, X_0, X_1, X_2, X_3\} = \{X(s, t), X(s - 2, t), X(s, t - 2), X(s, t + 2), X(s + 2, t)\}$, where $X$ can be a sample, the map index, or the similarity measure. The missing $G$ pixel at coordinates $(s, t)$, denoted by $G_c$, is interpolated according to Eq. (9). Note that Eqs. (11) and (12) must be applied beforehand. The entire $G$ plane is filled up after this stage is applied to all the missing $G$ pixels.

3.2 Inter RB-1 (C-Stage)

Let $\zeta = \{(s - 1, t - 1), (s, t - 1), (s - 1, t + 1), (s + 1, t), (s + 1, t + 1)\}$ in Fig. 7 (c). The missing $R$ pixel at coordinates $(s, t)$ is interpolated according to Eq. (10). Similarly, interpolate the missing $B$ pixels with $k = 2$. The half population of both $R$ and $B$ planes are filled up after this stage is applied to all the relevant pixels.

3.3 Inter RB-2 (D-Stage)

Let $\zeta = \{(s - 1, t), (s, t - 1), (s, t + 1), (s + 1, t)\} in Fig. 7 (d). The
missing $R$ pixel at coordinates $(s, t)$ is interpolated according to Eq. (10). Similarly, interpolate the missing $B$ pixels with $k = 2$. The whole population of both $R$ and $B$ planes are filled up after this stage is applied to all the relevant pixels.

4. Experiments

In the experiments, we measure the performance of the state-of-the-art CFA interpolations algorithms: bi-linear, method in [9], method in [11], DAF-CDM [13], and the proposed method with $M = N = 8$ and $d = 4$ configuration, and compare them quantitatively. We use three objective quality criteria: i) the mean absolute error (MAE), ii) the mean square error (MSE), and iii) the normalized color difference (NCD) defined in Eq. (13).

$$MAE = \frac{1}{3ST} \sum_{k=0}^{S-1} \sum_{s=0}^{T-1} |o(s, t) - x(s, t)|$$

$$MSE = \frac{1}{3ST} \sum_{k=0}^{S-1} \sum_{s=0}^{T-1} (o(s, t) - x(s, t))^2$$

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

where $o(s, t) = [o(s, t)_0, o(s, t)_1, o(s, t)_2]$ and $x(s, t) = [x(s, t)_0, x(s, t)_1, x(s, t)_2]$ denote the RGB vectors at co-ordinated $(s, t)$ of original image and reconstructed image (both images are of extension $S$ by $T$), respectively. Also $o'(s, t) = [o'(s, t)_0, o'(s, t)_1, o'(s, t)_2]$ and $x'(s, t) = [x'(s, t)_0, x'(s, t)_1, x'(s, t)_2]$ represent the RGB vectors $o(s, t)$ and $x(s, t)$ in the CIE LUV color space [15] with the white point D65.

We use 24 test images shown in Fig. 8, which are provided by Kodak. The input CFA data are captured from the original test images according to the Bayer CFA pattern shown in Fig. 1. Then, respective CFA interpolation algorithm is applied to the input CFA data.

Table 1 summarizes the simulation results, in which the best scores per metric for each image is represented in bold. We observe several tendencies in the results: i) The method in [9] works well for the images with gentle or weak color

| Image Number | MAE | MSE | NCD | MAE | MSE | NCD | MAE | MSE | NCD |
|--------------|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| 1            | 6.41| 150.16 | 0.0936 | 2.73 | 26.44 | 0.0420 | 2.66 | 25.11 | 0.0413 | 2.52 | 22.99 | 0.0384 |
| 2            | 2.46 | 29.33 | 0.0411 | 1.33 | 7.01 | 0.0245 | 1.36 | 6.92 | 0.0249 | 1.30 | 6.63 | 0.0503 |
| 3            | 1.94 | 22.27 | 0.0292 | 1.03 | 5.56 | 0.0175 | 1.10 | 5.73 | 0.0195 | 1.01 | 5.36 | 0.0181 |
| 4            | 2.44 | 26.19 | 0.0141 | 1.36 | 7.13 | 0.0261 | 1.39 | 7.25 | 0.0266 | 1.35 | 6.94 | 0.0256 |
| 5            | 5.54 | 135.30 | 0.1116 | 2.17 | 20.16 | 0.0538 | 2.11 | 18.78 | 0.0520 | 2.21 | 20.04 | 0.0533 |
| 6            | 4.74 | 102.56 | 0.0624 | 2.04 | 20.24 | 0.0285 | 1.97 | 18.49 | 0.0269 | 1.91 | 18.71 | 0.0264 |
| 7            | 2.17 | 28.34 | 0.0362 | 1.06 | 5.15 | 0.0194 | 1.08 | 5.13 | 0.0197 | 1.13 | 5.16 | 0.0205 |
| 8            | 7.86 | 269.96 | 0.1115 | 2.85 | 36.76 | 0.0450 | 2.81 | 35.63 | 0.0443 | 3.13 | 37.83 | 0.0471 |
| 9            | 2.52 | 35.20 | 0.0326 | 1.27 | 6.17 | 0.0174 | 1.32 | 6.32 | 0.0180 | 1.22 | 5.53 | 0.0177 |
| 10           | 2.48 | 35.45 | 0.0337 | 1.23 | 6.29 | 0.0183 | 1.28 | 6.16 | 0.0192 | 1.17 | 5.76 | 0.0174 |
| 11           | 3.85 | 74.69 | 0.0716 | 1.73 | 14.78 | 0.0356 | 1.69 | 14.02 | 0.0349 | 1.70 | 14.25 | 0.0354 |
| 12           | 2.33 | 88.27 | 0.0246 | 1.14 | 5.30 | 0.0130 | 1.21 | 5.55 | 0.0139 | 1.11 | 5.00 | 0.0127 |
| 13           | 8.27 | 255.85 | 0.1238 | 4.16 | 65.69 | 0.0696 | 4.11 | 64.35 | 0.0682 | 4.41 | 68.73 | 0.0719 |
| 14           | 4.18 | 72.67 | 0.0738 | 1.99 | 17.03 | 0.0402 | 1.98 | 16.88 | 0.0401 | 2.03 | 17.21 | 0.0406 |
| 15           | 2.45 | 36.87 | 0.0470 | 1.44 | 10.44 | 0.0310 | 1.49 | 10.64 | 0.0320 | 1.42 | 10.09 | 0.0299 |
| 16           | 3.12 | 46.11 | 0.0550 | 1.79 | 8.83 | 0.0256 | 1.35 | 8.06 | 0.0247 | 1.31 | 7.62 | 0.0239 |
| 17           | 2.54 | 38.49 | 0.0606 | 1.31 | 8.90 | 0.0371 | 1.32 | 8.79 | 0.0380 | 1.24 | 7.64 | 0.0354 |
| 18           | 4.49 | 98.74 | 0.0947 | 2.36 | 25.92 | 0.0369 | 2.35 | 25.33 | 0.0366 | 2.42 | 28.03 | 0.0571 |
| 19           | 4.17 | 95.48 | 0.0625 | 1.74 | 11.78 | 0.0295 | 1.76 | 11.47 | 0.0297 | 1.70 | 11.01 | 0.0282 |
| 20           | 2.36 | 28.82 | 0.0347 | 1.29 | 9.09 | 0.0201 | 1.28 | 8.74 | 0.0199 | 1.24 | 8.27 | 0.0189 |
| 21           | 4.09 | 87.54 | 0.0629 | 2.03 | 18.40 | 0.0336 | 2.05 | 17.92 | 0.0337 | 1.89 | 16.26 | 0.0308 |
| 22           | 3.40 | 55.60 | 0.0503 | 1.87 | 14.19 | 0.0301 | 1.89 | 14.04 | 0.0302 | 1.81 | 13.30 | 0.0289 |
| 23           | 1.69 | 19.48 | 0.0238 | 1.00 | 4.04 | 0.0161 | 1.03 | 4.17 | 0.0166 | 1.05 | 4.38 | 0.0171 |
| 24           | 4.86 | 140.89 | 0.0699 | 2.46 | 44.10 | 0.0388 | 2.47 | 44.98 | 0.0391 | 2.59 | 48.15 | 0.0403 |

Note: The best score per measure per image is represented in bold.
transitions such as the test images #7 and #23. ii) The results of the method in [11] are synchronized with those of the method in [9]. iii) The DAF-CDM [13] doesn’t work well for the images with steep or complicated color transitions such as the test images #8 and #13. iv) The proposed scheme yields the best score with over 85% probability (63/72, 72 = 24 images x 3 metrics). v) In PSNR (peak signal to noise ratio) wise, the proposed scheme outperforms the other methods by approximately 1.3 dB on average. It can be concluded that the proposed method outperforms the other algorithms under test in terms of the objective image qualities.

We also evaluate the subjective qualities of the reconstructed images. In general, visual appearance varies more likely on fine details such as edges (than stationary regions) depending on the CFA interpolation algorithm. Thus, a portion of the test image #19 (named Lighthouse) that contains such details is chosen as test sub-image, which is depicted as Fig. 9 (a). In Fig. 9, the reconstructed images by five algorithms under test are shown as well. We observe in Fig. 9: i) The bilinear method and DAF-CDM [13] suffer pixel-wise noises all over the entire regions. These noises make straight lines jaggy. ii) The proposed method reconstructs the fine details overall. iii) Both the methods in [9] and [11] produce noticeable line-wise artifacts that appear as horizontal short lines across the vertical lines whereas they well reconstruct the other regions. iv) The bilinear method and the method in [11] tend to smear the fine details. The visual comparison reveals that the proposed method produces less artifact while reproducing the fine details when compared with the other methods under test.

Table 2 shows the normalized computations required to interpolate one missing pixel with regard to primitive arithmetic operations. Then, we compare the proposed method with the state-of-the-art algorithms tested in the experiment above. Note that Table 2 shows the computations in $M = N = 8$ configuration for the proposed algorithm, and that $3/MN$ divisions are added and instead $3/MN$ shifts are subtracted for the configurations other than $M = N = 4, 8, 16$ (more precisely $2^p$, where $p$ is a natural number). The numbers in Table 2 indicate that the proposed algorithm requires less computational resources than the DAF-CDM, and that the proposal is as inexpensive as the methods in [9] and [11]. It is noteworthy that the DAF-CDM necessitates multiplication and division that is one of the most resource hungry operations whereas the other schemes under test (including the proposed method with $M = N = 8$ configuration) are free from multiplication and division.

5. Conclusion

This paper described a practical CFA interpolation technique that uses a local map called unified geometry map (UGM). We confirmed through the simulations that the proposed algorithm outperformed the state-of-the-art technologies in terms of objective quality measures. It was also con-
firmed that the proposed algorithm is comparable to, or even less expensive than the state-of-the-art technologies with regard to the resource requirements. Exploration of the CFA interpolation with multi-bit UGM is worth further investigation.

References

[1] B.E. Bayer, “Color imaging array,” U.S. Patent 3 971 065, July 1976.
[2] B. Gunturk, Y. Altunbasak, and R. Mersereau, “Color plane interpolation using alternating projections,” IEEE Trans. Image Process., vol.11, no.9, pp.997–1013, Sept. 2002.
[3] X. Li, “Demosaicing by successive approximation,” IEEE Trans. Image Process., vol.14, no.3, pp.370–379, March 2005.
[4] C.-Y. Su, “Highly effective iterative demosaicing using weighted-edge and color-difference interpolations,” IEEE Trans. Consum. Electron., vol.52, no.2, pp.639–645, Nov. 2006.
[5] R.H. Hibbard, “Apparatus and method for adaptively interpolating a full color image utilizing luminance gradients,” U.S. Patent 5, 382, 976, Jan. 1995.
[6] C.A. Laroche and M.A. Prescott, “Apparatus and method for adaptively interpolating a full color image utilizing chrominance gradients,” U.S. Patent 5, 373, 322, Dec. 1994.
[7] J.E. Adams and J.F. Hamilton Jr., “Adaptive color plane interpolation in single color electronic camera,” U.S. Patent 5, 506, 619, April 1996.
[8] J.E. Adams Jr., “Design of practical color filter array interpolation algorithms for digital cameras,” Proc. SPIE, vol.3028, pp.117–125, Feb. 1997.
[9] J.F. Hamilton Jr. and J.E. Adams, “Adaptive color plane interpolation in single sensor color electronic camera,” U.S. Patent 5, 629, 734, May 1997.
[10] T. Kuno and H. Sugiura, “Practical color filter array interpolation with constrained color correlation,” IEEE Trans. Consum. Electron., vol.52, no.3, pp.896–903, Aug. 2006.
[11] T. Kuno and H. Sugiura, “Practical color filter array interpolation part 2 with non-linear filter,” IEEE Trans. Consum. Electron., vol.52, no.4, pp.1409–1417, Nov. 2006.
[12] Y. Itoh, “CFA interpolation using unified geometry map,” Proc. FIT2008, vol.3, RI-002, pp.5–8, Sept. 2008.
[13] R. Lukac and K.N. Plataniotis, “Data adaptive filters for demosaicking: A framework,” IEEE Trans. Consum. Electron., vol.51, no.2, pp.560–570, May 2005.
[14] Y. Itoh, “Detail preserving noise filtering for compressed video,” IEICE Trans. Commun., vol.E79-B, no.10, pp.1459–1466, Oct. 1996.
[15] G. Sharma and H.J. Trussell, “Digital color imaging,” IEEE Trans. Image Process., vol.6, no.7, pp.901–932, July 1997.