IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received July 12, 2018, accepted August 13, 2018, date of publication August 31, 2018, date of current version September 21, 2018.

Digital Object Identifier 10.1109/ACCESS.2018.2868052

Decision-Based Demosaicking Algorithm Using Bayesian Theorem

DAEJUN PARK AND JECHANG JEONG, (Member, IEEE)
Department of Electronics and Computer Engineering, Hanyang University, Seoul 04763, South Korea
Corresponding author: Jechang Jeong (jjeong@hanyang.ac.kr)

ABSTRACT In this paper, we introduce a new color filter array interpolation based on edge map prediction using the Bayesian theorem. The edge information obtained at the position of the green component sampled in the quincunx grid using the Bayer pattern is used to predict the edge of the red and blue component positions distributed in the rectangular grid. In this process, the edge distribution of the entire image, local area, and nearest neighbors is analyzed to determine the smooth/edge characteristics stochastically. If the pixels at the red and blue positions are determined to be edge pixels, the edge direction is inherited from the edge pixels of the local region. By combining a new decision technique with directional weighted interpolation in the green plane reconstruction process, the interpolation accuracy can be improved compared with the existing demosaicking algorithms. After interpolating the green channel, the color difference between the interpolated green channel and the red/blue channel is used to refine the reconstructed green plane. The other color planes can then be interpolated using this refined green plane. Experimental results show that our algorithm improves both the objective and subjective image qualities compared with the conventional state-of-the-art demosaicking algorithms.

INDEX TERMS Color filter array interpolation, demosaicking, decision-based interpolation, edge map prediction, Bayesian theorem, Bayer pattern.

I. INTRODUCTION

To represent a digital color image, three color channels are required for each pixel [1]. Three sensors should be used to obtain information about each channel. However, because of the portability and cost of a camera, a single sensor-based digital imaging device is used in most camera types. A single image sensor obtains information about one color per pixel through a color filter array (CFA). The most common CFA is the Bayer CFA pattern [2], where the green (G) component is sampled on a quincunx grid, and the red (R) and blue (B) components are sampled on rectangular grids, as shown in Fig. 1. Thus, 50% of the CFA image is the G component, with 25% each for the R/B components. A color information estimation process is performed to obtain the remaining unknown color information that does not exist at each pixel position. This interpolation technique is called CFA interpolation or demosaicking [3].

The G component is sampled twice as much as the R/B components. This is because the human visual system (HVS) [4] has the highest sensitivity at medium wavelengths, which occupy the G color part. Therefore, a luminance image without aliasing is first generated through G component interpolation at the R/B positions. Because the interpolated G image is used to generate the remaining R/B components, the prediction of the G component is important. The R/B components are predicted based on the assumption that the color difference signals are locally constant. If necessary, the color components of the pixel are modified using a post-refinement process. The overall process described above is a general CFA interpolation process.

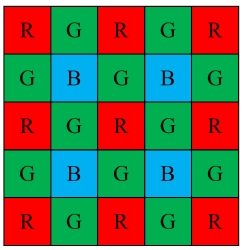

FIGURE 1. Example of a 5 × 5 Bayer filter [2].

Various demosaicking algorithms based on the Bayer pattern have been proposed over the past few decades [7]–[16]. In order to accurately and effectively estimate non-existent

components, it has been widely assumed that the color differences are constant in a small region [5]. This assumption, however, produces bad results for regions where drastic changes exist such as the presence of edges. Accordingly, many algorithms have been developed that first perform edge prediction and then perform interpolation along the edge direction [6].

Adams and Hamilton [7] proposed the adaptive color plane interpolation (ACPI) algorithm to predict the direction of the edge. ACPI performs adaptive interpolation through a classifier using the Laplacian second-order value, gradient value, and color difference bias value. In [8], an effective directional weighted interpolation method (DWI) was proposed to reconstruct a full color image using spectral correlation in a specific direction. The authors use a Sobel filter to approximate the directional gradient, and the absolute value of gradients in order to improve the computation speed. Effective color interpolation (ECI) [9] performs interpolation using the color difference between the G and R/B color planes to obtain a completely reconstructed image. Using the property that the HVS is sensitive to chromatic changes rather than luminance changes in low spatial frequency regions, the authors suggested a demosaicking system with low complexity. Through the above algorithms, the necessary theories in the demosaicking process have been established, and the improved algorithms have been proposed.

Because the ACPI method determines the interpolation direction based on the horizontal and vertical gradient approximations, it can result in unsatisfying results in regions with textures or thin objects. Therefore, studies for predicting more accurate edges have continued. Chung and Chan [10] proposed the local variance of color differences (VCD) as a homogeneity criterion in order to apply a more reliable method to edge direction selection. Su and Kao [11] proposed effective demosaicking using subband correlation (EDUSC) utilizing the property of a wavelet transform to classify edge pixels. In [12], the authors proposed an edge strength filter (ESF) that provides local, orientation-free luminance transition information and applied it to the demosaicking problem.

Algorithms have been developed to eliminate extreme color artifacts that occur because ECI does not consider edge directionality in the interpolation process. An effective demosaicking algorithm based on the edge property (EDAEP) [13] was proposed to improve the performance of the existing ECI algorithm and eliminate color artifacts. The authors effectively detected the edges of an image and generated more accurate weights for image interpolation based on the horizontal or vertical directionality. Color demosaicking with directional filtering and weighting (CDDFW) [14] was proposed based on a combination of directional filtering, an a posteriori decision method, and enhanced ECI. The authors analyzed the advantages and limitations of the conventional demosaicking methods and tried to eliminate the errors that are still obvious.

Since existing algorithms have been specialized for specific image dataset with high spectral correlations [5], demosaicking algorithms using natural images [25], [26] as test sequences have also been developed. In [15], the authors presented the successive refinement demosaicking algorithm (SRD), which is based on DWI and gradient inverse-weighted filtering. The DWI-based refinement method proposed in SRD is effective in reducing interpolation artifacts, with a slight increase in complexity and better results than existing interpolation enhancing approaches. An effective demosaicking technique based on the Taylor series and adaptive directional selection (TAD) was proposed for adaptive direction and weight selection to achieve high efficiency in real time [16]. The authors used a large local window and a small local window together to increase the accuracy of the interpolation, and experimentally selected a function to determine the weight. In recent years, researchers have been able to acquire improved images by proposing a learning-based demosaicking technique [17]–[19].

The conventional edge sensing-based demosaicking algorithms commonly perform edge detection for the interpolation of the G components at the R/B pixel positions. However, their performances may be inaccurate because they simply detect edges by comparing the differences between neighboring pixels. In addition, because the edge distribution of the surrounding pixels is not considered, an outlier occurs in some cases. Although incorrectly determining that a smooth pixel is an edge pixel has no significant effect on the demosaicking result, if an edge pixel is determined to be a smooth pixel, this has a large effect on the demosaicking result.

In this paper, we propose an edge map prediction using Bayesian theorem to obtain accurate edge information used in the interpolation process. First, edge detection is performed on existing G pixels to determine smooth/edge characteristics information. Next, Bayesian theorem-based smooth/edge decision probabilities at R/B pixel positions are calculated by analyzing the edge distribution of the entire image, local area, and nearest neighbor pixels. Finally, when the R/B pixel position is determined as an edge, the edge direction is inherited from the surrounding G pixels having the edge characteristics. Then, CFA interpolation is performed according to the generated edge map.

The remainder of this paper is organized as follows. In section II, we describe the Bayesian theorem-based edge map prediction. Section III describes the demosaicking algorithm used in this paper. Experimental results using test images are presented in section IV. Finally, some conclusions are presented in section V.

II. BAYESIAN THEOREM-BASED EDGE MAP PREDICTION

As shown in Fig. 2, the proposed decision-based demosaicking system is divided into two processes, an edge map generation process and a CFA interpolation process using the edge map. The conventional edge determination method directly determines the smooth/edge characteristics and edge

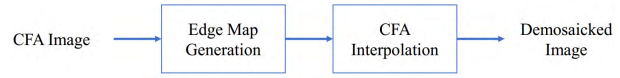

FIGURE 2. Flowchart of proposed demosaicking system.

direction using the difference between neighboring pixels at the R/B positions. Moreover, the prediction accuracy is low because the edge distribution of the surrounding pixels is not considered. In this section, we propose a new edge map generation method, as shown in Fig. 3. The strength of the edge prediction process using Bayesian theorem is that the edge information of the G pixels can be utilized in the R/B pixel positions.

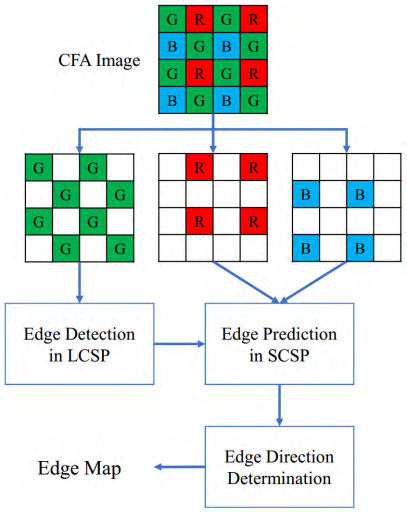

FIGURE 3. Flowchart of proposed edge map generation process.

A. EDGE DETECTION

The edge determination method proposed in this paper predicts the smooth/edge characteristics of the R/B pixel positions using the edge information of the original G component. Therefore, it is necessary to determine whether the original G pixel is a smooth pixel or an edge pixel through edge detection at the G component position, and to determine the directionality. In this process, the gradients (sum of absolute differences) in the horizontal and vertical directions are calculated.

In the Bayer pattern, G pixels are fully sampled in diagonal directions [20]. However, in the R/B pixel positions, the G pixels are distributed in the cardinal directions, and the edge information of the diagonal directions is meaningless. In addition, since HVS is more sensitive to horizontal and vertical edges than diagonal edges [21], we only consider cardinal directions in edge map generation and G plane interpolation process.

As shown in Fig. 4, we use eight surrounding G pixels to determine the edge information at the G5 location. The gradient changes in the horizontal and vertical directions are obtained using (1) and (2). It can be said that a sharp

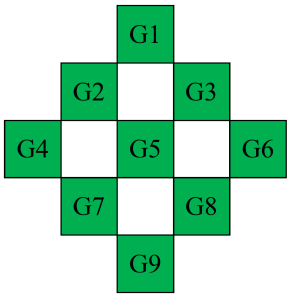

FIGURE 4. Central G5 and eight surrounding G pixels used in the edge detection process.

edge exists when one of the horizontal and vertical gradient changes is relatively large compared to the other value.

$$H5 = |G4 - G5| + |G5 - G6| + |G2 - G3| + |G7 - G8|, \tag{1}$$

$$V5 = |G1 - G5| + |G5 - G9| + |G2 - G7| + |G3 - G8|. \tag{2}$$

The edge determination is performed by comparing the ratio values of $H5$ and $V5$. If the larger value of $H5/V5$ or $V5/H5$ is greater than the predetermined threshold T, the G5 position is determined to be a strong edge. In the opposite case, the G5 position is classified as a smooth pixel. If the larger value is $H5/V5$, it is said to have a vertical edge because the horizontal gradient change is larger. Conversely, if the $V5/H5$ value is larger, it is said to have a horizontal edge.

B. BAYESIAN THEOREM-BASED EDGE MAP PREDICTION

The Bayesian theorem is used to calculate the smooth/edge determination probability when interpolating the G component at the R/B pixel positions. In this process, the edge distributions of the entire image and local region, and the large cross-shaped pattern (LCSP) G component edge distribution are synthetically considered. In other words, a greater number of edges for the entire image or adjacent region will lead to a higher probability that the characteristics of the G component to be interpolated is an edge. The edge prediction of the small cross-shaped pattern (SCSP) is performed by analyzing the edge distribution of the LCSP G components. The shape of these two patterns is shown in Fig. 5.

The probability of the smooth/edge characteristics at the R/B positions based on the Bayesian theorem when using the edge map of the original G component is obtained by

$$p(Type|Case) = \frac{p(Case|Type) \cdot p(Type)}{p(Case)}. \tag{3}$$

where *Case* represents the edge distribution of the nearest neighbors and is shown as sixteen cases in Fig. 6. *Type* indicates the smooth/edge characteristics of the corresponding pixel position. For example, $p(Type = Smooth|Case = 1)$ indicates the probability that the type of the current pixel is a smooth when all of the closest neighbors are smooth pixels.

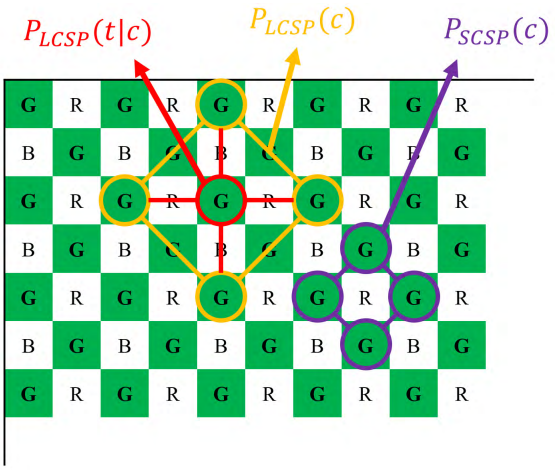

FIGURE 5. Pixels corresponding to LCSP and SCSP used in the edge map prediction process.

LCSP

for all G pixels (raster scan)
  if nearest G pixels' pattern (LCSP) == c_i
    N_{c_i}++
    if current pixel type == Smooth
      $N_{S|c_i}$++
      N_S++
    else
      $N_{E|c_i}$++
      N_E++
    end
  end
end

$$p_{LCSP}(c_i) = \frac{N_{c_i}}{\sum_i N_{c_i}}$$

$$p_{LCSP}(t|c_i) = \frac{N_{t|c_i}}{N_{S|c_i} + N_{E|c_i}}$$

$$p_{LCSP}(t) = \frac{N_t}{N_S + N_E}$$

where $i = 1, 2, \cdots, 16$ and $t \in \{S, E\}$.

SCSP

for all R/B pixels (raster scan)
  if nearest G pixels' pattern (SCSP) == c_i
    n_{c_i}++
  end
end

$$p_{SCSP}(c_i) = \frac{n_{c_i}}{\sum_i n_{c_i}}$$

Edge Map Prediction

$$p_{LCSP}(c_i|t) = \frac{p_{LCSP}(t|c_i) \cdot p_{LCSP}(c_i)}{p_{LCSP}(t)}$$

for all R/B pixels (raster scan)
  if nearest G pixels' pattern (SCSP) == c_i
    calculate $p_{SCSP}(t)$ in a 5×5 region

$$p_{SCSP}(t|c_i) = \frac{p_{LCSP}(t|c_i) \cdot p_{LCSP}(c_i) \cdot p_{SCSP}(t)}{p_{LCSP}(t) \cdot p_{SCSP}(c_i)}$$

    if $p_{SCSP}(Smooth|c_i) > p_{SCSP}(Edge|c_i)$
      Set as a Smooth pixel
    else
      Set as an Edge pixel
    end
  end
end

FIGURE 7. Pseudo code for the edge prediction of nonexistent G pixels at R/B pixel locations.

A
B X C
D

Cross-Shaped Pattern

Case	A	B	C	D	Case	A	B	C	D
1	S	S	S	S	9	E	E	S	E
2	E	E	E	E	10	E	E	E	S
3	E	S	S	S	11	E	E	S	S
4	S	E	S	S	12	E	S	E	S
5	S	S	E	S	13	S	S	E	E
6	S	S	S	E	14	S	E	S	E
7	S	E	E	E	15	E	S	S	E
8	E	S	E	E	16	S	E	E	S

FIGURE 6. Selection of CSP pixels and sixteen cases of characteristics distributions (S : Smooth, E : Edge).

$p(Case|Type)$ is the likelihood that plays the most important role in finding the probability of the smooth/edge determination. $p(Type)$ represents the distribution of smooth or edge pixels in a given region, and $p(Case)$ represents the occupancy rate of a specific case in a given cross grid. For simplicity, we use the lowercase alphabet to represent $p(t|c)$, $p(c|t)$, $p(t)$, and $p(c)$.

In order to determine whether the pixel position to be interpolated is a smooth or an edge, all of the information on the right side of (3) is needed. Our final goal is to find the smooth/edge decision probability of the R/B pixel locations in SCSP. Thus, we use (4) below.

$$p_{SCSP}(t|c) = \frac{p_{SCSP}(c|t) \cdot p_{SCSP}(t)}{p_{SCSP}(c)}. \tag{4}$$

where $p_{SCSP}(c)$ is the case probability according to the surrounding edge distribution in the SCSP. The edge distribution of a given region can be expressed by $p_{SCSP}(t)$. In this paper, we use the edge distribution of the original G pixels in a 5×5 region. Because $p_{SCSP}(c|t)$ is an unknown likelihood, we must derive the value using the information obtained from the LCSP edge analysis. Because images usually have geometric duality properties which means the consistency of geometric structure across resolutions, high-resolution parameters can be estimated from low-resolution samples [22]– [24]. Therefore, we can derive $p_{SCSP}(c|t)$ by analyzing the LCSP edge map. To obtain the unknown likelihood $p_{SCSP}(c|t)$, we use (5).

$$p_{SCSP}(c|t) \approx p_{LCSP}(c|t) = \frac{p_{LCSP}(t|c) \cdot p_{LCSP}(c)}{p_{LCSP}(t)}. \tag{5}$$

To obtain $p_{LCSP}(c|t)$, we calculate the $p_{LCSP}(t|c)$, $p_{LCSP}(c)$, and $p_{LCSP}(t)$ probability tables with a raster scan over the entire image as the pseudo code in Fig. 7. In the LCSP grid, the frequency of cases is increased according to the surrounding edge distribution and divided by the sum of all the cases to obtain $p_{LCSP}(c)$, as shown in (6).

$$p_{LCSP}(c_i) = \frac{N_{c_i}}{\sum_i N_{c_i}}. \tag{6}$$

where N_{c_i} is the number of occurrences of case c_i and $i = 1, 2, \cdots, 16$. The above procedure can be used to calculate $p_{LCSP}(t|c)$ according to the smooth/edge type of the center pixel. In addition, we use the smooth/edge ratio of the G pixels over the entire image as the probability of $p_{LCSP}(t)$, as shown in (7).

$$\begin{cases} p_{LCSP}(t|c_i) = \dfrac{N_{t|c_i}}{N_{S|c_i} + N_{E|c_i}}, \\ p_{LCSP}(t) = \dfrac{N_t}{N_S + N_E}. \end{cases} \tag{7}$$

where $t|c_i$ means that the case of CSP is c_i and the type of the center pixel is $t \in \{Smooth, Edge\}$. $N_{t|c_i}$ is the number of occurrences of the center pixel type according to the case of surrounding pixels of the CSP grid within the image. And N_t is the number of occurrences of the center pixel type t. Using this information, a 16×2 $p_{LCSP}(c|t)$ probability table can be calculated for sixteen cases and two pixel types.

The method of obtaining $p_{LCSP}(c)$ from the LCSP grid is applied to the SCSP grid to obtain $p_{SCSP}(c)$. Then, the $p_{SCSP}(t|c)$ of the R/B pixel positions in (4) is obtained using the above information. If $p_{SCSP}(Smooth|c)$ is greater

than $p_{SCSP}(Edge|c)$, the corresponding pixel is classified as a smooth pixel.

C. EDGE DIRECTION DETERMINATION

When the pixels at the R/B positions are determined to be edge pixels, the edge direction is inherited from the edge pixels of the specific region. As shown in Fig. 8, the edge direction of the X position is determined by using the edge information of twelve G pixels. The directionality of a plurality of edge pixels is defined as the directionality of X.

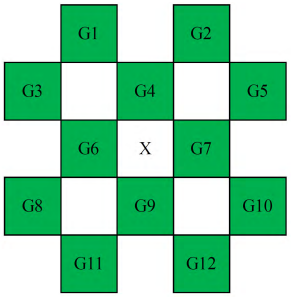

FIGURE 8. Central X and original G pixels in 5 × 5 region used in the edge direction determination process.

III. COLOR FILTER ARRAY INTERPOLATION

The CFA interpolation process shown in Fig. 9 is used to interpolate missing color components. Using the edge information obtained in section II, the interpolation is performed using the gradient inverse value as a weight according to the pixel characteristics. The color difference model is used to reduce color artifacts in the high-frequency region. This is because the HVS is sensitive to the high-frequency region, and the existing color ratio model produces bad results in the edge region.

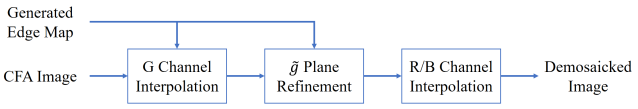

FIGURE 9. Flowchart of CFA interpolation method.

The assumption used in the color difference model is that the three channels (R, G, and B) have similar high-frequency components. In addition, because the G channel has the property of a luminance component, the high-frequency map of the G component can replace the high-frequency map of the image. Therefore, the G channel with low aliasing error is preferentially interpolated in the R/B pixel locations using the predicted G component-based edge map. We then perform a refinement step to obtain an accurate G plane. The missing R/B components are interpolated to complete the demosaicking process.

In the interpolation process, the original color components are expressed as R, G, and B using capital letters, and the components used in the intermediate interpolation process are represented by r, g, and b in lower case. The final interpolated components are represented by $\hat{r}$, $\hat{g}$, and $\hat{b}$.

A. GREEN CHANNEL INTERPOLATION

Because the HVS is more sensitive to the G component, it has twice the distribution in the CFA image as the R/B components. In other words, it can be said that the G component contains the structural information of the image, and the interpolation of the G component affects the interpolation of the other color components. Therefore, the G channel interpolation is performed first using the DWI algorithm.

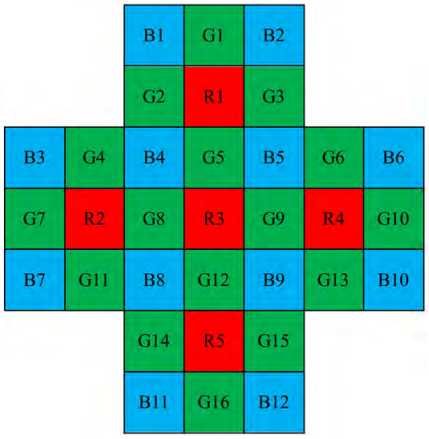

FIGURE 10. Reference block used to interpolate $\tilde{g}_{R3}$ of R3.

Fig. 10 shows the reference block used to interpolate the G component at the R position. Because R/B have the same sampling position characteristics, only the G component interpolation at the R position is considered in this paper.

A relatively simple demosaicking method is applied considering the complexity of the proposed edge map prediction process using the Bayesian theorem. To interpolate the $\tilde{g}_{R3}$ of R3, we use the first-order derivative in the four directions of north (N), south (S), west (W), and east (E) using the neighboring pixels. Directional information for these four directions is shown in (8).

$$\begin{cases} g_N = G5 + (R3 - R1)/2, \\ g_S = G12 + (R3 - R5)/2, \\ g_W = G8 + (R3 - R2)/2, \\ g_E = G9 + (R3 - R4)/2. \end{cases} \tag{8}$$

An appropriate weight is assigned to a predicted value in each direction in order to perform the correct interpolation according to the image details. For this purpose, we use (9) to obtain the color gradients using the color differences in the corresponding direction.

$$\begin{cases} \nabla_N = |G5 - G1| + |G8 - G2| + |G9 - G3| \\ \quad +|B4 - B1| + |B5 - B2| + |R3 - R1| + \varepsilon, \\ \nabla_S = |G12 - G16| + |G8 - G14| + |G9 - G15| \\ \quad +|B8 - B11| + |B9 - B12| + |R3 - R5| + \varepsilon, \\ \nabla_W = |G8 - G7| + |G5 - G4| + |G12 - G11| \\ \quad +|B4 - B3| + |B8 - B7| + |R3 - R2| + \varepsilon, \\ \nabla_E = |G9 - G10| + |G5 - G6| + |G12 - G13| \\ \quad +|B5 - B6| + |B9 - B10| + |R3 - R4| + \varepsilon. \end{cases} \tag{9}$$

where ε is added so that the denominator of the weight does not become zero. The weight related to the predicted value in each direction is determined by the reciprocal of the color gradient, as shown in (10). A smoother area results in a smaller color gradient and larger value for the weight. In a region where a color change is large and color artifacts can occur, the color gradient is large, and the value of the weight is small. Therefore, the initial interpolated $\tilde{g}_{R3}$ is obtained by applying the weights according to the pixel characteristics.

$$w_N = \frac{1}{\nabla_N}, w_S = \frac{1}{\nabla_S}, w_W = \frac{1}{\nabla_W}, w_E = \frac{1}{\nabla_E}. \quad (10)$$

Finally, the G channel interpolation using the weighted average method is performed according to the smooth/edge characteristics and edge direction of R3. If the characteristics of the R3 pixel is a smooth, the interpolation method of (11) is applied. The distribution of the characteristics of all the surrounding pixels is taken into consideration without specific directionality.

$$\tilde{g}_{R3} = \frac{g_N \cdot w_N + g_S \cdot w_S + g_W \cdot w_W + g_E \cdot w_E}{w_N + w_S + w_W + w_E}. \quad (11)$$

If the characteristics of the R3 pixel is an edge and shows vertical or horizontal directionality, directional interpolation is performed using (12) or (13). This process makes it possible to avoid interpolation across the edge.

$$\tilde{g}_{R3} = \frac{g_N \cdot w_N + g_S \cdot w_S}{w_N + w_S}, \quad (12)$$

$$\tilde{g}_{R3} = \frac{g_W \cdot w_W + g_E \cdot w_E}{w_W + w_E}. \quad (13)$$

The initial interpolated G plane can be obtained using the above procedure. To improve the prediction accuracy of the R/B components, it is essential to update the G plane using a refinement process.

B. GREEN PLANE REFINEMENT PROCESS

After interpolating the G channel according to the pixel characteristics, the color difference between the interpolated G channel and the R/B channel is used to refine $\tilde{g}$ (see the reference area in Fig. 10). In this process, we use the four nearest G components with the highest color correlation, and calculate the color gradients to obtain the weight values in four directions. We use (14) to obtain the color gradients.

$$\begin{cases} \delta_N = |G5 - \tilde{g}_{R3}| + |R1 - R3| + \varepsilon, \\ \delta_S = |G12 - \tilde{g}_{R3}| + |R5 - R3| + \varepsilon, \\ \delta_W = |G8 - \tilde{g}_{R3}| + |R2 - R3| + \varepsilon, \\ \delta_E = |G9 - \tilde{g}_{R3}| + |R4 - R3| + \varepsilon. \end{cases} \quad (14)$$

The refinement process is applied by specifying an appropriate weight according to the characteristics of each direction. We use the inverse of the color gradients in the weight calculation process, as shown in (15).

$$w'_N = \frac{1}{\delta_N}, \quad w'_S = \frac{1}{\delta_S}, \; w'_W = \frac{1}{\delta_W}, \; w'_E = \frac{1}{\delta_E}. \quad (15)$$

For the pixel positions classified as a smooth in the edge map prediction process, the final G component is obtained through a refinement process considering all directions using (16), as shown at the bottom of this page.

For pixel positions classified as a vertical or horizontal edge, we apply (17) or (18) to refine the interpolated G pixels.

$$\hat{g}_{R3} = R3 + \frac{\tilde{g}_{R3} - R3}{2} + \frac{(G5 - R3) \cdot w'_N + (G12 - R3) \cdot w'_S}{2 \cdot (w'_N + w'_S)}, \quad (17)$$

$$\hat{g}_{R3} = R3 + \frac{\tilde{g}_{R3} - R3}{2} + \frac{(G8 - R3) \cdot w'_W + (G9 - R3) \cdot w'_E}{2 \cdot (w'_W + w'_E)}. \quad (18)$$

The remaining process is predicting the R/B components using the color difference plane. The R/B components are first predicted at the G position. Next, the R (B) component prediction at the B (R) position is performed.

C. R/B CHANNEL INTERPOLATION

After the $\hat{g}$ plane is obtained through the interpolation and refinement process, the interpolation of the R/B components is performed. Because R/B pixels are strongly associated with G pixels, interpolation is performed using the color difference model.

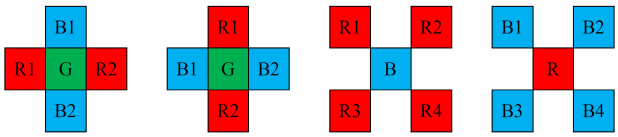

FIGURE 11. The reference blocks used to interpolate $\hat{r}$ and $\hat{b}$.

Fig. 11 shows the reference blocks used in the R/B interpolation. First, the R/B components at the G pixel positions are interpolated. In the case of G pixels with the shapes shown in Fig. 11(a) and 11(b), the $\hat{r}_G/\hat{b}_G$ components are interpolated using (19) and (20), respectively.

$$\hat{r}_G = G + \frac{R1 - \hat{g}_{R1} + R2 - \hat{g}_{R2}}{2}, \quad (19)$$

$$\hat{b}_G = G + \frac{B1 - \hat{g}_{B1} + B2 - \hat{g}_{B2}}{2}. \quad (20)$$

Finally, for those shown in Fig. 11(c) and 11(d), the $\hat{r}_B/\hat{b}_R$ components are obtained using (21) and (22), respectively.

$$\hat{g}_{R3} = R3 + \frac{\tilde{g}_{R3} - R3}{2} + \frac{(G5 - R3) \cdot w'_N + (G12 - R3) \cdot w'_S + (G8 - R3) \cdot w'_W + (G9 - R3) \cdot w'_E}{2 \cdot (w'_N + w'_S + w'_W + w'_E)}. \quad (16)$$

It is now possible to fully reconstruct the color image.

$$\hat{r}_B = \hat{g}_B + \frac{1}{4}\sum_{i=1}^{4}(Ri - \hat{g}_{Ri}), \tag{21}$$

$$\hat{b}_R = \hat{g}_R + \frac{1}{4}\sum_{i=1}^{4}(Bi - \hat{g}_{Bi}). \tag{22}$$

IV. EXPERIMENTAL RESULTS

This section shows how we verified the superiority of the proposed algorithm through comparisons with existing algorithms (VCD [10], EDUSC [11], ESF [12], EDAEP [13], CDDFW [14], SRD [15], and TAD [16]). Post-processing was performed using median filtering to remove demosaicking artifacts. We used the McMaster (McM) dataset (with a size of 500×500) [25] to compare and evaluate the performances of the algorithms, as shown in Fig. 12. In addition, Laurent Condat's (LC) Image Database [26] (Fig. 13) is used to evaluate the subjective performance of the proposed CFA interpolation system. Since the Kodak images used as test images in the existing demosaicking process are scanned version from film-based photos, they are not suitable as test images for the CFA interpolation method [5]. Also, because the spectral correlations of these test images are significantly higher than typical color images, they yield biased experimental results in the design of the demosaicking system [15]. Therefore, the experiments are conducted using McM and LC datasets with lower spectral correlations and close to images captured by color sensors.

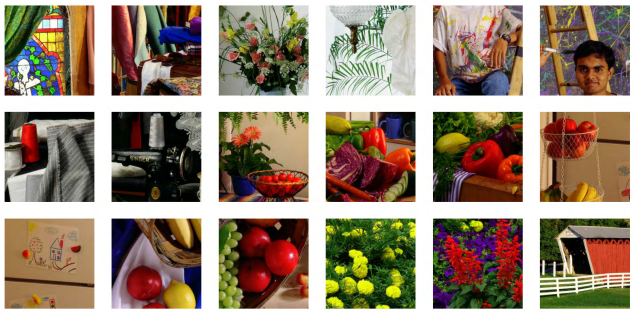

FIGURE 12. McM dataset (size of 500 × 500 in TIFF format) [25].

The objective performance was assessed in terms of the color peak signal-to-noise ratio (CPSNR) [3], feature similarity index measure for color images (FSIMc) [27], S-CIELAB ΔE^* [28], zipper effect ratio (ZER) [8], and CPU time. We also investigated the performance of the proposed algorithm using a subjective image quality comparison. Our experiment was performed on an Intel Core i7-8700K CPU @ 3.70GHz with MATLAB R2018a. In the edge detection process, the threshold value $T = 2$ was used. Experiments with the reference algorithms used the MATLAB code published by the authors.

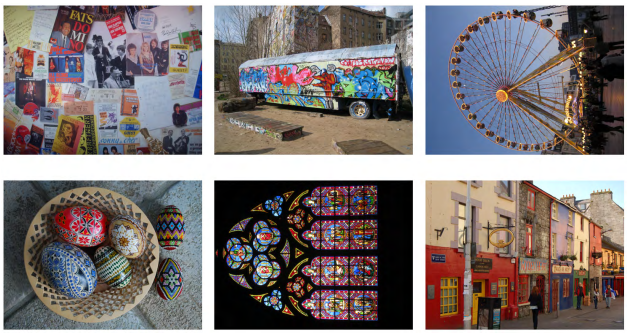

FIGURE 13. LC dataset (size of 720 × 540 or 540 × 720 in TIFF format) [26].

A. OBJECTIVE PERFORMANCE ANALYSIS

The objective performance results were evaluated using five quality assessment measures and are listed in Tables 1-5. Because the algorithm proposed in this paper focuses on the edge decision process, it uses a relatively simple interpolation technique compared to the reference algorithms. Nevertheless, the experimental results showed the superiority of the proposed algorithm.

TABLE 1. CPSNR results (dB) of existing algorithms and proposed algorithm for McM dataset.

McM	VCD	EDUSC	ESF	EDAEP	CDDFW	SRD	TAD	Proposed
1	34.7274	33.8434	33.4539	34.7038	34.7518	35.4884	35.2267	35.4626
2	38.1614	37.2405	36.6402	38.2215	38.0609	38.6416	38.6893	38.7291
3	36.9338	36.7852	36.3614	36.9606	37.1964	37.9064	37.9856	37.5584
4	38.8667	38.2388	37.5863	38.8473	38.4622	41.5554	40.9735	40.4647
5	37.9844	37.0668	36.5230	37.9571	37.9297	39.0850	38.8327	39.1318
6	39.1931	37.3721	36.3321	39.1383	39.0690	40.8839	40.1347	40.5366
7	40.4042	42.2561	42.9796	39.8758	41.1090	38.7777	40.0095	38.8562
8	41.8291	41.5351	41.2262	41.4634	42.0450	41.5530	42.1557	41.5865
9	39.3045	38.1986	37.6856	39.4681	39.2656	40.8822	40.5789	40.9789
10	40.6224	39.1611	38.2706	40.6405	40.4358	41.2861	41.3181	41.3167
11	41.4689	40.1076	39.3910	41.6193	41.5993	42.4427	42.3172	42.5565
12	41.1848	39.9513	39.8139	41.4017	41.4790	42.2955	42.2612	42.1093
13	42.8222	41.9454	41.2179	42.8563	42.8558	44.1837	43.7293	44.0957
14	41.3530	40.6065	39.7822	41.4177	41.4400	42.1335	42.0664	42.2147
15	41.2369	40.4078	39.7147	41.3366	41.1530	42.1534	41.9738	42.2521
16	37.1661	36.1010	35.5167	37.3835	37.4388	38.3035	38.0932	38.5012
17	36.1897	34.7277	34.3469	36.2680	35.9721	38.1487	37.0603	37.3514
18	38.2171	37.1221	36.6941	38.2825	38.2715	39.3187	39.0854	39.3885
Average	39.3148	38.4815	37.9187	39.3246	39.3631	40.2800	40.1384	40.1717

Table 1 shows that the CPSNR performance of the proposed algorithm was the best for half of the test images. The average CPSNR result was the best for SRD, but because CPSNR is an evaluation measure that does not take human visual characteristics into account, it must be accompanied by an evaluation using HVS-based measures.

Tables 2 and 3 present the results of experiments using the FSIMc and S-CIELAB ΔE^* metrics based on the HVS to determine whether the structural characteristics of the images match. When the FSIMc is closer to 1.0, the demosaicked image is more similar to the original image. Because S-CIELAB ΔE^* compares perceptual differences, the performance of the demosaicking algorithm could be accurately compared. The experimental results showed the superior performance of the proposed algorithm.

Table 4 lists experimental results showing how the zipper effect occurs in each algorithm in the demosaicking process.

TABLE 2. FSIMc results of existing algorithms and proposed algorithm for McM dataset.

McM	VCD	EDUSC	ESF	EDAEP	CDDFW	SRD	TAD	Proposed
1	0.99217	0.98922	0.98578	0.99251	0.99303	0.99378	0.99406	0.99389
2	0.99643	0.99537	0.99402	0.99657	0.99670	0.99678	0.99709	0.99704
3	0.99695	0.99702	0.99577	0.99718	0.99741	0.99722	0.99775	0.99719
4	0.99813	0.99799	0.99698	0.99837	0.99833	0.99889	0.99914	0.99873
5	0.99685	0.99536	0.99354	0.99697	0.99720	0.99801	0.99795	0.99805
6	0.99820	0.99676	0.99430	0.99824	0.99838	0.99876	0.99880	0.99879
7	0.99817	0.99878	0.99853	0.99808	0.99818	0.99747	0.99799	0.99744
8	0.99781	0.99774	0.99688	0.99802	0.99797	0.99769	0.99805	0.99791
9	0.99735	0.99655	0.99465	0.99754	0.99768	0.99787	0.99809	0.99813
10	0.99836	0.99791	0.99697	0.99839	0.99856	0.99869	0.99884	0.99870
11	0.99824	0.99790	0.99681	0.99838	0.99857	0.99864	0.99876	0.99859
12	0.99826	0.99790	0.99717	0.99837	0.99836	0.99873	0.99881	0.99871
13	0.99797	0.99733	0.99590	0.99804	0.99827	0.99870	0.99863	0.99881
14	0.99809	0.99760	0.99671	0.99814	0.99836	0.99857	0.99859	0.99869
15	0.99790	0.99769	0.99693	0.99796	0.99814	0.99824	0.99831	0.99836
16	0.99714	0.99597	0.99288	0.99742	0.99765	0.99794	0.99811	0.99803
17	0.99531	0.99325	0.98985	0.99571	0.99604	0.99710	0.99675	0.99633
18	0.99747	0.99686	0.99561	0.99760	0.99768	0.99782	0.99815	0.99799
Average	0.99727	0.99651	0.99496	0.99742	0.99758	0.99783	0.99799	0.99786

TABLE 3. S-CIELAB ΔE^* results of existing algorithms and proposed algorithm for McM dataset.

McM	VCD	EDUSC	ESF	EDAEP	CDDFW	SRD	TAD	Proposed
1	2.9735	4.5958	4.8418	2.9348	3.0127	2.6149	2.6001	2.5028
2	1.2421	1.8616	1.9255	1.2423	1.3306	1.1544	1.1068	1.1098
3	2.2113	2.1941	2.5957	2.1648	2.1962	1.5177	1.5906	1.6950
4	1.3397	1.5828	1.9685	1.3922	1.5403	0.7518	0.8761	1.0103
5	1.5261	1.9273	2.2893	1.5084	1.5405	1.2539	1.2530	1.2146
6	1.3262	2.0753	2.8210	1.3217	1.3410	1.0726	1.0983	1.0326
7	1.0803	0.9459	0.9914	1.1378	1.0351	1.1842	1.1029	1.2583
8	0.6564	0.7973	0.8633	0.7015	0.6561	0.6543	0.5927	0.6224
9	1.3476	1.7343	1.9236	1.3227	1.4093	0.9853	1.0149	0.9947
10	0.9897	1.3014	1.5768	0.9853	1.0138	0.8774	0.8495	0.8774
11	0.7573	1.0182	1.1574	0.7459	0.7673	0.6515	0.6529	0.6457
12	1.0070	1.3499	1.5455	0.9398	1.0456	1.0052	0.9352	0.8830
13	0.7030	0.8585	0.8945	0.6872	0.7113	0.6035	0.5978	0.5937
14	0.8002	0.9643	1.0862	0.7813	0.7778	0.7059	0.6918	0.6967
15	0.8553	0.9917	1.1354	0.8258	0.8496	0.7760	0.7561	0.7341
16	1.8666	2.8537	3.4715	1.8599	1.9080	1.6257	1.6163	1.4792
17	2.3130	3.6456	3.9487	2.2848	2.5013	1.7390	1.9701	1.8355
18	1.4655	2.0581	2.1665	1.4813	1.5793	1.4397	1.2954	1.3155
Average	1.3590	1.8198	2.0668	1.3510	1.4009	1.1452	1.1445	1.1390

TABLE 4. ZER results of existing algorithms and proposed algorithm for McM dataset.

McM	VCD	EDUSC	ESF	EDAEP	CDDFW	SRD	TAD	Proposed
1	0.3120	0.3031	0.3815	0.3255	0.3055	0.2828	0.3092	0.2893
2	0.1999	0.1838	0.2358	0.1995	0.1868	0.1655	0.1789	0.1691
3	0.1201	0.1208	0.1573	0.1159	0.1244	0.1358	0.1312	0.1146
4	0.0169	0.0135	0.0292	0.0147	0.0295	0.0641	0.0315	0.0180
5	0.1014	0.0786	0.1079	0.0997	0.0952	0.0811	0.0845	0.0796
6	0.0628	0.0661	0.0932	0.0609	0.0508	0.0406	0.0458	0.0402
7	0.0217	0.0167	0.0240	0.0206	0.0214	0.0291	0.0217	0.0316
8	0.0383	0.0338	0.0380	0.0378	0.0344	0.0447	0.0385	0.0445
9	0.1532	0.1540	0.2026	0.1510	0.1472	0.1177	0.1374	0.1122
10	0.1223	0.1278	0.1611	0.1242	0.1165	0.0937	0.1070	0.0951
11	0.1222	0.1444	0.1753	0.1271	0.1204	0.0961	0.1091	0.0857
12	0.0593	0.0487	0.0853	0.0610	0.0507	0.0506	0.0497	0.0426
13	0.0268	0.0246	0.0387	0.0261	0.0230	0.0140	0.0194	0.0160
14	0.1822	0.1732	0.2166	0.1795	0.1761	0.1533	0.1663	0.1485
15	0.1848	0.1782	0.2284	0.1822	0.1865	0.1524	0.1682	0.1489
16	0.3385	0.4328	0.4796	0.3576	0.3644	0.2923	0.3439	0.2499
17	0.3747	0.3804	0.4442	0.3798	0.3659	0.3190	0.3519	0.3002
18	0.2623	0.2596	0.3123	0.2504	0.2723	0.1997	0.2465	0.1719
Average	0.1500	0.1522	0.1895	0.1507	0.1484	0.1296	0.1411	0.1199

TABLE 5. CPU time results (s) of existing algorithms and proposed algorithm for McM dataset.

McM	VCD	EDUSC	ESF	EDAEP	CDDFW	SRD	TAD	Proposed
Average	0.2159	0.5577	1.6835	0.1254	0.3503	1.0847	0.0879	0.4075

The zipper effect, which refers to an on-off pattern in regions with saturated colors, is one of the most common artifacts in CFA interpolation. It can be seen that a lower ZER is associated with fewer zipper artifacts in the image. The proposed algorithm had the lowest ZER average as a result of the correct prediction of the edge map. Finally, Table 5 lists the average CPU execution times.

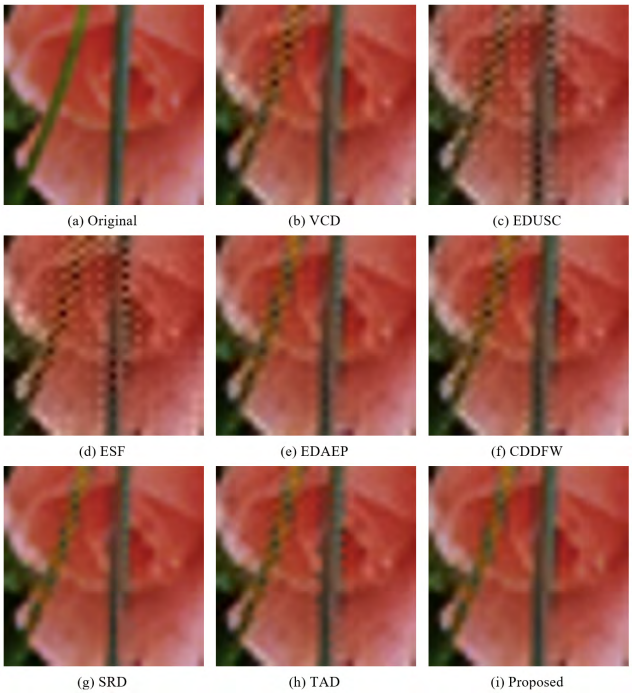

FIGURE 14. Comparison of (a) original image and result of the (b) VCD, (c) EDUSC, (d) ESF, (e) EDAEP, (f) CDDFW, (g) SRD, (h) TAD, and (i) Proposed methods (McM #3).

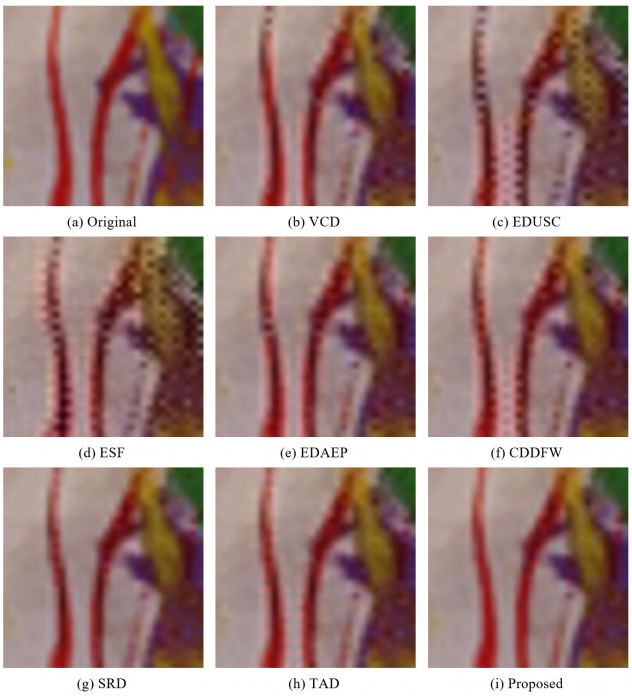

FIGURE 15. Comparison of (a) original image and result of the (b) VCD, (c) EDUSC, (d) ESF, (e) EDAEP, (f) CDDFW, (g) SRD, (h) TAD, and (i) Proposed methods (McM #5).

B. SUBJECTIVE PERFORMANCE ANALYSIS

Figs. 14-17 shows a subjective image comparison of the demosaicked images of McM images (#3, #5, #8, and #13).

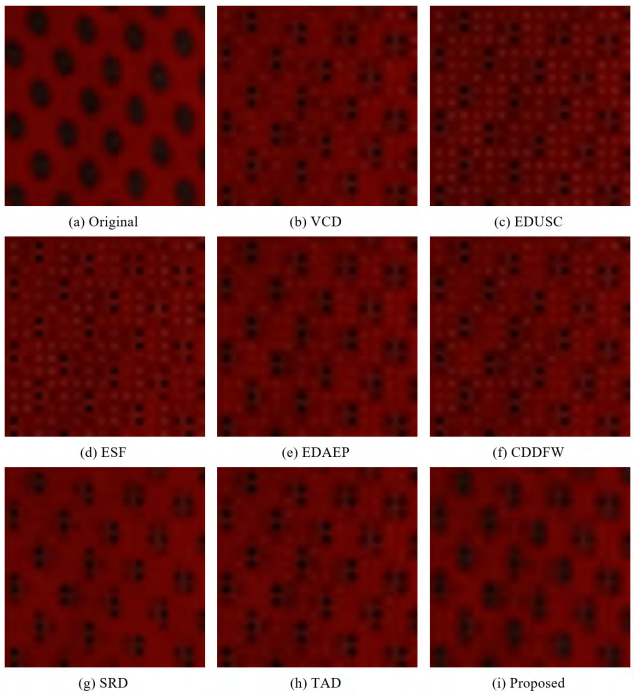

FIGURE 16. Comparison of (a) original image and result of the (b) VCD, (c) EDUSC, (d) ESF, (e) EDAEP, (f) CDDFW, (g) SRD, (h) TAD, and (i) Proposed methods (McM #8).

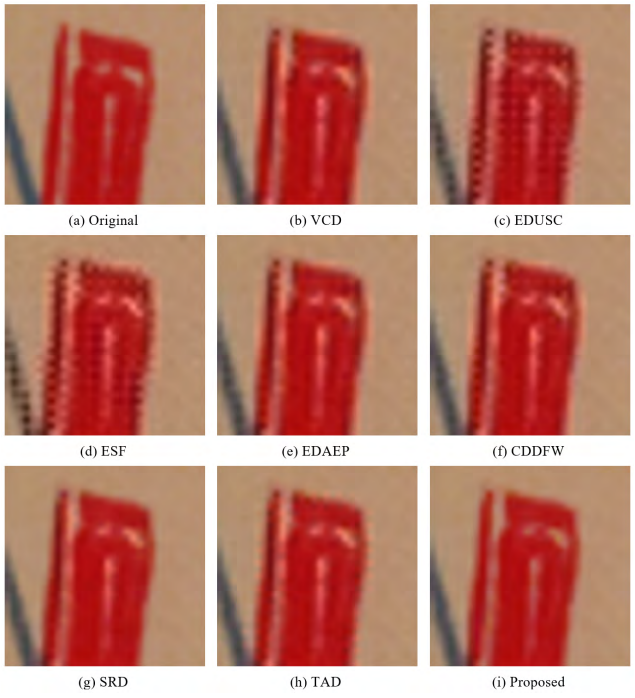

FIGURE 17. Comparison of (a) original image and result of the (b) VCD, (c) EDUSC, (d) ESF, (e) EDAEP, (f) CDDFW, (g) SRD, (h) TAD, and (i) Proposed methods (McM #13).

FIGURE 18. Comparison of (a) original image and result of the (b) VCD, (c) EDUSC, (d) ESF, (e) EDAEP, (f) CDDFW, (g) SRD, (h) TAD, and (i) Proposed methods (LC #1).

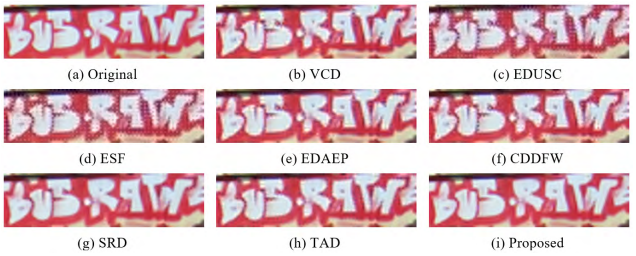

FIGURE 19. Comparison of (a) original image and result of the (b) VCD, (c) EDUSC, (d) ESF, (e) EDAEP, (f) CDDFW, (g) SRD, (h) TAD, and (i) Proposed methods (LC #2).

FIGURE 20. Comparison of (a) original image and result of the (b) VCD, (c) EDUSC, (d) ESF, (e) EDAEP, (f) CDDFW, (g) SRD, (h) TAD, and (i) Proposed methods (LC #3).

We compared the performances of the algorithms by comparing the quantities of zipper and color artifacts that occurred in the images. As can be seen from the plant stem in Fig. 14, we can see distorted parts in the resulting images of most algorithms. However, the result image of the proposed algorithm shows that the artifacts are less. As shown in Fig. 15, the color artifacts on the red line are also noticeably severe.

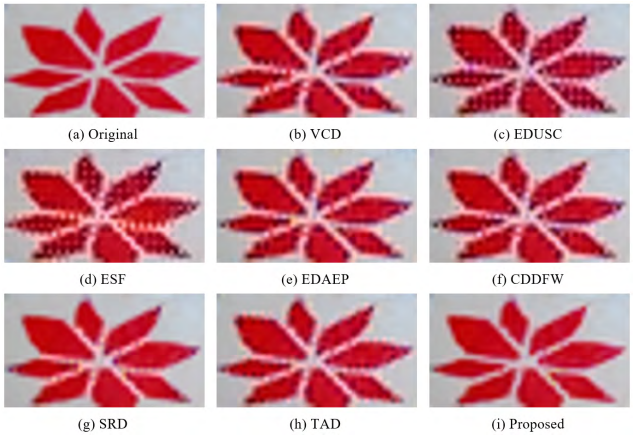

FIGURE 21. Comparison of (a) original image and result of the (b) VCD, (c) EDUSC, (d) ESF, (e) EDAEP, (f) CDDFW, (g) SRD, (h) TAD, and (i) Proposed methods (LC #4).

FIGURE 22. Comparison of (a) original image and result of the (b) VCD, (c) EDUSC, (d) ESF, (e) EDAEP, (f) CDDFW, (g) SRD, (h) TAD, and (i) Proposed methods (LC #5).

This is due to the wrong orientation in the edge determination process. As shown in Fig. 16, the superiority of the proposed algorithm can be seen even for images with repeated patterns. The pattern disappears in all the algorithms except the proposed algorithm. Moreover, the image quality degradation is severe at the color saturated boundary. Although the thin line color on the left side of Fig. 17 is actually red, it is purple in the results of all of the algorithms except the proposed algorithm.

FIGURE 23. Comparison of (a) original image and result of the (b) VCD, (c) EDUSC, (d) ESF, (e) EDAEP, (f) CDDFW, (g) SRD, (h) TAD, and (i) Proposed methods (LC #6).

Through previous experiments using the McM dataset, the superiority of the proposed algorithm is fully demonstrated. However, in order to show that the proposed algorithm is not biased in the McM dataset, additional subjective image quality evaluation was performed. Figs. 18-23 presents a subjective image comparison of the demosaicked images of LC dataset. Overall, the resulting images of the SRD, TAD, and proposed algorithms show relatively good results. On the other hand, the results of EDUSC and ESF algorithms show noticeable degradation in image quality due to zipper artifacts.

Fig. 18 shows that the image quality degradation due to zipper artifacts also occurs in VCD, EDAEP, and CDDFW demosaicked images. The proposed algorithm partially generates color artifacts that occur in the cardinal direction-based algorithm, but there are no on-off pattern artifacts. Figs. 19(h) and 20(h) show that artifacts occur at the boundary of the object because TAD uses a large kernel window during the interpolation process. In Figs. 21(g) and 22(g), the SRD result images can be partially observed to have demosaicking artifacts. Notice that the resulting images in Fig. 23 undergo color artifacts. However, if we look at Fig. 23(i), which is the result of the proposed algorithm, it can be seen that no zipper artifacts due to misjudgment of the edge direction occur.

As can be seen from the various experimental results, the accuracy of the Bayesian theorem-based edge map prediction of the proposed demosaicking algorithm is high. Based on the well-predicted edge map, the interpolation process can generate images closer to the original compared to other existing algorithms.

V. CONCLUSION

In this paper, we proposed a new demosaicking algorithm based on edge map prediction using the Bayesian theorem. Compared to the existing edge sensing algorithms, accurate edge prediction was possible and effectively eliminated color artifacts due to interpolation in the wrong direction. Our proposed method was shown to have superior performance compared to conventional algorithms through objective and subjective performance analyses.

REFERENCES

[1] R. Kimmel, "Demosaicing: Image reconstruction from color CCD samples," *IEEE Trans. Image Process.*, vol. 8, no. 9, pp. 1221–1228, Sep. 1999.
[2] B. E. Bayer, "Color imaging array," U.S. Patent 3 971 065, Jul. 20, 1976.
[3] D. Menon and G. Calvagno, "Color image demosaicking: An overview," *Signal Process., Image Commun.*, vol. 26, nos. 8–9, pp. 518–533, Oct. 2011.
[4] D. Alleysson, S. Süsstrunk, and J. Hérault, "Linear demosaicing inspired by the human visual system," *IEEE Trans. Image Process.*, vol. 14, no. 4, pp. 439–449, Apr. 2005.
[5] X. Li, B. Gunturk, and L. Zhang, "Image demosaicing: A systematic survey," *Proc. SPIE*, vol. 6822, p. 68221J, Jan. 2008, doi: 10.1117/12.766768.
[6] B. K. Gunturk, J. Glotzbach, Y. Altunbasak, R. W. Schafer, and R. M. Mersereau, "Demosaicking: Color filter array interpolation," *IEEE Signal Process. Mag.*, vol. 22, no. 1, pp. 44–54, Jan. 2005.
[7] J. E. Adams and J. F. John, "Adaptive color plane interpolation in single color electronic camera," U.S. Patent 5 506 619, Apr. 9, 1996.
[8] W. Lu and Y.-P. Tan, "Color filter array demosaicking: New method and performance measures," *IEEE Trans. Image Process.*, vol. 12, no. 10, pp. 1194–1210, Oct. 2003.
[9] S.-C. Pei and I.-K. Tam, "Effective color interpolation in CCD color filter arrays using signal correlation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 6, pp. 503–513, Jun. 2003.
[10] K.-H. Chung and Y.-H. Chan, "Color demosaicing using variance of color differences," *IEEE Trans. Image Process.*, vol. 15, no. 10, pp. 2944–2955, Oct. 2006.
[11] C.-Y. Su and W.-C. Kao, "Effective demosaicing using subband correlation," *IEEE Trans. Consum. Electron.*, vol. 55, no. 1, pp. 199–204, Feb. 2009.
[12] I. Pekkucuksen and Y. Altunbasak, "Edge strength filter based color filter array interpolation," *IEEE Trans. Image Process.*, vol. 21, no. 1, pp. 393–397, Jan. 2012.
[13] W.-J. Chen and P.-Y. Chang, "Effective demosaicking algorithm based on edge property for color filter arrays," *Digital Signal Process.*, vol. 22, no. 1, pp. 163–169, Jan. 2012.
[14] Z. Dengwen, S. Xiaoliu, and D. Weiming, "Colour demosaicking with directional filtering and weighting," *IET Image Process.*, vol. 6, no. 8, pp. 1084–1092, Nov. 2012.
[15] X. Chen, L. He, G. Jeon, and J. Jeong, "Color filter array interpolation by successive refinement over color channels using gradient inverse-weighted filtering," *Opt. Commun.*, vol. 318, pp. 189–198, May 2014.
[16] J. Wang, J. Wu, Z. Wu, and G. Jeon, "Taylor series and adaptive directional selection for real time demosaicking," *Displays*, vol. 45, pp. 14–25, Dec. 2016.
[17] F.-L. He, Y.-C. F. Wang, and K.-L. Hua, "Self-learning approach to color demosaicking via support vector regression," in *Proc. 19th IEEE ICIP*, Orlando, FL, USA, Sep./Oct. 2012, pp. 2765–2768.
[18] K.-L. Hua, S. C. Hidayati, F.-L. He, C.-P. Wei, and Y.-C. F. Wang, "Context-aware joint dictionary learning for color image demosaicking," *J. Vis. Commun. Image Represent.*, vol. 38, pp. 230–245, Jul. 2016.
[19] D. S. Tan, W.-Y. Chen, and K.-L. Hua, "DeepDemosaicking: Adaptive image demosaicking via multiple deep fully convolutional networks," *IEEE Trans. Image Process.*, vol. 27, no. 5, pp. 2408–2419, May 2018.
[20] L. Chang and Y.-P. Tan, "Hybrid color filter array demosaicking for effective artifact suppression," *J. Electron. Imag.*, vol. 15, no. 1, p. 013003, Jan. 2006.
[21] R. Ramanath, W. E. Snyder, Y. Yoo, and M. S. Drew, "Color image processing pipeline," *IEEE Signal Process. Mag.*, vol. 22, no. 1, pp. 34–43, Jan. 2005.
[22] X. Li and M. T. Orchard, "New edge-directed interpolation," *IEEE Trans. Image Process.*, vol. 10, no. 10, pp. 1521–1527, Oct. 2001.
[23] K. W. Hung and W. C. Siu, "Robust soft-decision interpolation using weighted least squares," *IEEE Trans. Image Process*, vol. 21, no. 3, pp. 1061–1069, Mar. 2012.
[24] C. Ren, X. He, Q. Teng, Y. Wu, and T. Q. Nguyen, "Single image super-resolution using local geometric duality and non-local similarity," *IEEE Trans. Image Process.*, vol. 25, no. 5, pp. 2168–2183, May 2016.
[25] L. Zhang, X. Wu, A. Buades, and X. Li, "Color demosaicking by local directional interpolation and nonlocal adaptive thresholding," *J. Electron. Imag.*, vol. 20, no. 2, p. 023016, Apr. 2011.
[26] L. Condat. (2010). *Image Available*. [Online]. Available: http://www.gipsa-lab.grenoble-inp.fr/~laurent.condat/imagebase.html
[27] L. Zhang, L. Zhang, X. Mou, and D. Zhang, "FSIM: A feature similarity index for image quality assessment," *IEEE Trans. Image Process.*, vol. 20, no. 8, pp. 2378–2386, Aug. 2011.
[28] X. Zhang and B. Wandell, "A spatial extension of CIELAB for digital color-image reproduction," *J. Soc. Inf. Display*, vol. 5, no. 1, pp. 61–63, Mar. 1997.

DAEJUN PARK received the B.S. degree from the Department of Electronics and Computer Engineering, Hanyang University, Seoul, South Korea, in 2011, where he is currently pursuing the Ph.D. degree with the Department of Electronics Engineering. His research interests include digital signal processing, image interpolation, image processing, high-dynamic-range imaging, and HDR/WCG video compression.

JECHANG JEONG received the B.S. degree in electronic engineering from Seoul National University, Seoul, South Korea, in 1980, the M.S. degree in electrical and electronic engineering from the Korea Advanced Institute of Science and Technology, Daejeon, South Korea, in 1982, the Ph.D. degree in electrical engineering from the University of Michigan, Ann Arbor, in 1990. From 1982 to 1986, he was with the Korean Broadcasting System, where he helped develop teletext systems. From 1990 to 1991, he was a Post-Doctoral Research Associate with the University of Michigan, Ann Arbor, where he helped develop various signal processing algorithms. From 1991 to 1995, he was with Samsung Electronics Company, South Korea, where he was involved in the development of HDTV, digital broadcasting receivers, and other multimedia systems. Since 1995, he has been conducting research at Hanyang University, Seoul. His research interests include digital signal processing, digital communication, and image and audio compression for HDTV and multimedia applications. He has published numerous technical papers. He received the Scientist of the Month Award from the Ministry of Science and Technology of Korea in 1998. He was a recipient of the 2007 IEEE Chester Sall Award and the 2008 ETRI Journal Paper Award. He was also honored with a government commendation in 1998 from the Ministry of Information and Communication of Korea.