DIRECTIONAL COLOR FILTER ARRAY INTERPOLATION BASED ON MULTISCALE COLOR GRADIENTS

Ibrahim Pekkucuksen, Yucel Altunbasak

Center for Signal and Image Processing, Georgia Institute of Technology, Atlanta, GA, 30308, USA

ABSTRACT

Single sensor digital cameras capture one color value for every pixel location. The remaining two color channel values need to be estimated to obtain a complete color image. This process is called demosaicing or Color Filter Array (CFA) interpolation. We propose a directional approach to the CFA interpolation problem that makes use of multiscale color gradients. The relationship between color gradients on different scales is used to generate signals in vertical and horizontal directions. We determine how much each direction should contribute to the green channel interpolation based on these signals. The proposed method is easy to implement since it is noniterative and threshold free. Experiments on test images show that it offers superior objective and subjective interpolation quality.

Index Terms— Demosaicing, color filter array interpolation, bayer color filter array, multiscale color gradient, directional interpolation

1. INTRODUCTION

Most digital cameras employ single sensor designs because using multiple sensors coupled with beam splitters for each pixel location is costly in hardware. This design choice necessitates the use of color filter arrays. The color channel layout on a color filter array determines which channel will be captured at each pixel location. Many different CFA layouts have been proposed but the Bayer CFA pattern is the most commonly used design [1]. The color channel layout for the Bayer CFA pattern is shown in Figure 1.

While it is possible to use simple linear interpolation techniques to complete the missing color channel information, such nonadaptive approaches lead to low quality output with false color artifacts and blurriness. In addition to the available spatial correlation, there is a spectral correlation between the color channels and any high quality CFA interpolation algorithm needs to take advantage of this information source somehow. Demosaicing solutions combining spatial adaptiveness and spectral correlation have been introduced early on. Hamilton et al. proposed adaptively interpolating the green channel in horizontal or vertical directions, or a combination of both, based on directional classifiers and thresholds [2]. The idea of using available red and blue channel pixels in initial green channel interpolation is borrowed by many subsequent methods. A possible area for improvement is to come up with better classifiers that can lead to a more accurate direction decision. Variance of color differences is used to make a hard interpolation direction decision in [3], while linear minimum mean-square error framework is employed to combine directional estimates in [4]. Another interesting approach is to interpolate the green channel in both directions and then to make a posteriori decision based on sum of gradients in each direction [5].

The demosaicing problem has been studied from many other angles. Glotzbach et al. proposed a frequency domain approach where they extracted high frequency components from the green channel and used them to improve red and blue channel interpolation [6]. Gunturk et al. used the strong spectral correlation between high frequency subbands to develop an alternating projections method [7]. A comprehensive list of demosaicing approaches, experimental results, and observations are presented in a recent survey paper [8].

The rest of the paper is organized as follows. Section 2 gives some background information and describes the proposed method in detail, section 3 presents experimental results, and section 4 gives a brief discussion.

2. PROPOSED ALGORITHM

2.1. Algorithm Background

We proposed a directional CFA interpolation method that uses color difference gradients in [9]. We have been looking for ways to improve its performance and examining color difference gradients on the pixel level was one of areas that we focused on. The horizontal color difference gradient on a red-green row is given as follows:

\[ D_{i,j}^H = |\hat{\Delta}_{i,j}^H - \hat{\Delta}_{i,j+1}^H | \] (1)

where a color difference estimate is given by:

\[ \tilde{\Delta}_{g,r}^H (i,j) = \begin{cases} \tilde{G}_{i,j}^H - R_{i,j}, & G \text{ is interpolated} \\ \tilde{G}_{i,j}^H - \tilde{R}_{i,j}, & R \text{ is interpolated} \end{cases} \] (2)
Fig. 1: Bayer CFA pattern.

and the pixel estimates themselves are calculated by:

$$\tilde{G}_{i,j} = \frac{G_{i,j-1} + G_{i,j+1}}{2} + \frac{2 \cdot R_{i,j} - R_{i,j-2} - R_{i,j+2}}{4}$$

$$\tilde{R}_{i,j} = \frac{R_{i,j-1} + R_{i,j+1}}{2} + \frac{2 \cdot G_{i,j} - G_{i,j-2} - G_{i,j+2}}{4}.$$  \hspace{1cm} (3)

We can write the color difference gradient (for a red target pixel) in terms of red and green pixels as follows:

$$D_{i,j}^H = \frac{1}{4} \left( \left| \frac{2 \cdot G_{i,j+1} + G_{i,j-3} + G_{i,j+1} - R_{i,j-2} - R_{i,j+2}}{2} \right| - \left| \frac{2 \cdot G_{i,j+1} + G_{i,j-1} + G_{i,j+3} - R_{i,j} + R_{i,j+2}}{2} \right| \right)$$

$$D_{i,j}^H = \frac{1}{4} \left( \left| \frac{2 \cdot G_{i,j+1} + G_{i,j-3} + G_{i,j+1} - R_{i,j-2} - R_{i,j+2}}{2} \right| - \left| \frac{2 \cdot G_{i,j+1} + G_{i,j-1} + G_{i,j+3} - R_{i,j} + R_{i,j+2}}{2} \right| \right)$$

$$D_{i,j}^V = \frac{1}{4} \left( \left| \frac{G_{i+1,j} - G_{i-1,j}}{2} - \frac{R_{i+2,j} - R_{i-2,j}}{4} \right| \right)$$

$$D_{i,j}^V = \frac{1}{4} \left( \left| \frac{G_{i+1,j} - G_{i-1,j}}{2} - \frac{R_{i+2,j} - R_{i-2,j}}{4} \right| \right)$$

Here, the first part of the equation is simply the green channel gradient and the second part is the red channel gradient on a larger scale with both parts normalized by the distance between their operands.

The whole equation checks whether both color gradients are in agreement with each other. They are likely to be close to each other in smooth regions and along edge structures while they are likely to be different across edges with sudden color changes. Thus, it can be used as a feature to combine directional estimates adaptively.

It should be noted that we can extend the same idea to larger scales. However, we need to keep in mind that the locality will get weaker with each additional scale. We can optimize the normalizing terms in the denominators to take this effect into account.

$$D_{i,j}^h = \frac{1}{4} \left( \left| \frac{G_{i,j+1} - G_{i,j-1}}{2} - \frac{R_{i,j+2} - R_{i,j-2}}{4} \right| \right)$$

$$D_{i,j}^v = \frac{1}{4} \left( \left| \frac{G_{i+1,j} - G_{i-1,j}}{2} - \frac{R_{i+2,j} - R_{i-2,j}}{4} \right| \right)$$

where the $N_i$ terms are the normalizers. The equations are similar for green-blue rows and green-blue columns.

### 2.2. Initial Green Channel Interpolation

The first step of the proposed algorithm is to interpolate the missing green channel pixels. We perform this interpolation adaptively using the multiscale color gradients equation derived above.

In addition to the horizontal pixel value and color difference estimations described in equations (2) and (3), vertical estimations are calculated by:

$$G_{i,j}^V = \frac{1}{4} \left( \left| \frac{G_{i-1,j} + G_{i+1,j}}{2} - \frac{2 \cdot R_{i-1,j} - R_{i+1,j}}{4} \right| \right)$$

$$R_{i,j}^V = \frac{1}{4} \left( \left| \frac{G_{i-1,j} + G_{i+1,j}}{2} - \frac{2 \cdot R_{i-1,j} - R_{i+1,j}}{4} \right| \right)$$

$$G_{i,j}^V = \frac{1}{4} \left( \left| \frac{G_{i-1,j} + G_{i+1,j}}{2} - \frac{2 \cdot R_{i-1,j} - R_{i+1,j}}{4} \right| \right)$$

$$R_{i,j}^V = \frac{1}{4} \left( \left| \frac{G_{i-1,j} + G_{i+1,j}}{2} - \frac{2 \cdot R_{i-1,j} - R_{i+1,j}}{4} \right| \right)$$

Next, we combine the directional color differences adaptively:

$$\hat{\Delta}_{g,r}(i,j) = \left[ w_V * f * \hat{\Delta}_{g,r}^V(i-1 : i+1,j) + w_H * \hat{\Delta}_{g,r}^H(i,j-1 : j+1) * f' \right] / w_T$$

$$\hat{\Delta}_{g,r}(i,j) = \left[ w_V * f * \hat{\Delta}_{g,r}^V(i-1 : i+1,j) + w_H * \hat{\Delta}_{g,r}^H(i,j-1 : j+1) * f' \right] / w_T$$
\[ w_T = w_V + w_H \]

\[ f = [1 \ 2 \ 1]/4. \quad (10) \]

The weights for horizontal and vertical directions \((w_H, w_V)\) are calculated by adding multiscale color gradients over a local window. For a window size of 5 by 5:

\[
\begin{align*}
    w_V &= 1/((\sum_{a=i-2}^{i+2} \sum_{b=j-2}^{j+2} D_{a,b}^v))^2 \\
    w_H &= 1/((\sum_{a=i-2}^{i+2} \sum_{b=j-2}^{j+2} D_{a,b}^h))^2 \quad (11)
\end{align*}
\]

The division operation can be avoided by defining the weights as the denominators and exchanging them (The ratio of \(1/a\) to \(1/b\) is equal to the ratio of \(b\) to \(a\) provided that both are nonzero).

### 2.3. Green Channel Update

After the initial green channel interpolation, we update the results using directional multiscale gradients again, except we evaluate north-south and east-west directions separately this time.

\[
\begin{align*}
    \tilde{\Delta}_{g,r}(i,j) &= \Delta_{g,r}(i,j) + (1 - w) + w_N \Delta_{g,r}(i-2,j) + w_S \Delta_{g,r}(i+2,j) + w_E \Delta_{g,r}(i,j-2) + w_W \Delta_{g,r}(i,j+2) * w/ w_T \\
    w_T &= w_N + w_S + w_E + w_W \quad (12)
\end{align*}
\]

Here, \(w\) is a number between 0 and 1 that determines how aggressive the update is. The weight for each direction \((w_N, w_S, w_E, w_W)\) is calculated by summing multiscale color gradients over a local window. Assuming a 3 by 5 window for horizontal and a 5 by 3 window for vertical components:

\[
\begin{align*}
    w_N &= 1/((\sum_{a=i-4}^{i} \sum_{b=j-1}^{j+1} D_{a,b}^v))^2 \\
    w_S &= 1/((\sum_{a=i}^{i+4} \sum_{b=j-1}^{j+1} D_{a,b}^v))^2 \\
    w_W &= 1/((\sum_{a=i-1}^{i+1} \sum_{b=j-4}^{j} D_{a,b}^v))^2 \\
    w_E &= 1/((\sum_{a=i-1}^{i+1} \sum_{b=j}^{j+4} D_{a,b}^v))^2 \quad (13)
\end{align*}
\]

Finally, the updated color difference estimate is added to the available target pixel to obtain the green channel estimate:

\[
\begin{align*}
    \tilde{G}(i,j) &= R(i,j) + \tilde{\Delta}_{g,r}(i,j) \\
    \tilde{G}(i,j) &= B(i,j) + \tilde{\Delta}_{g,b}(i,j) \quad (14)
\end{align*}
\]

### 2.4. Red and Blue Channel Interpolation

For red and blue channel interpolation, we keep the same approach that we employed in [9]. Red pixel values at blue locations and blue pixel values at red locations are interpolated using the filter that was proposed in [10]:

\[
\begin{align*}
    p_{rb} &= \begin{bmatrix}
        0 & 0 & -1 & 0 & -1 & 0 & 0 \\
        0 & 0 & 0 & 0 & 0 & 0 & 0 \\
        -1 & 0 & 10 & 0 & 10 & 0 & -1 \\
        0 & 0 & 0 & 0 & 0 & 0 & 0 \\
        -1 & 0 & 10 & 0 & 10 & 0 & -1 \\
        0 & 0 & 0 & 0 & 0 & 0 & 0 \\
        0 & -1 & 0 & -1 & 0 & 0 & 0
    \end{bmatrix} * \frac{1}{32}
\end{align*}
\]

\[
\begin{align*}
    \tilde{R}_{i,j} &= \tilde{G}_{i,j} - \Delta_{g,r}(i-3 : i+3, j-3 : j+3) \otimes p_{rb} \\
    \tilde{B}_{i,j} &= \tilde{G}_{i,j} - \Delta_{g,b}(i-3 : i+3, j-3 : j+3) \otimes p_{rb} \quad (15)
\end{align*}
\]

where \(\otimes\) denotes element-wise matrix multiplication and then summation of elements.

For red and blue pixels at green locations, we make use of the multiscale color gradients again. The horizontal and vertical estimations are combined adaptively using the directional weights \((w_H, w_V)\) defined in equation (11). The immediate vertical neighbors of a green pixel are either red or blue pixels. For the red pixel case the interpolation is carried out as follows:

\[
\begin{align*}
    \tilde{R}(i,j) &= G(i,j) - \frac{w_V \ast \tilde{G}_{i-1,j} - R_{i-1,j} + \tilde{G}_{i+1,j} - R_{i+1,j} + w_H \ast \tilde{G}_{i,j-1} - R_{i,j-1} + \tilde{G}_{i,j+1} - R_{i,j+1}}{2 \ast (w_V + w_H)} \\
    \tilde{B}(i,j) &= G(i,j) - \frac{w_V \ast \tilde{G}_{i-1,j} - B_{i-1,j} + \tilde{G}_{i+1,j} - B_{i+1,j} + w_H \ast \tilde{G}_{i,j-1} - B_{i,j-1} + \tilde{G}_{i,j+1} - B_{i,j+1}}{2 \ast (w_V + w_H)} \quad (16)
\end{align*}
\]

By the end of this step, all the missing values are estimated and the full color image is reconstructed.

### 3. EXPERIMENTAL RESULTS

We tested the proposed algorithm on the 12 image Kodak test set featured in [8]. The results in terms of CPSNR are compared to the three highest performing methods in a recent survey paper [8], and to the method that served as the starting point of the proposed algorithm [9]. These methods are, Gradient Based Threshold Free CFA (GBTF) [9], Local Polynomial Approximation (LPA) [10], Directional Linear Minimum Mean Square-Error Estimation (DLMMSE) [4], and
Table 1: Comparison of CPSNR values for different demosaicing methods.

<table>
<thead>
<tr>
<th>No.</th>
<th>VCD</th>
<th>DL</th>
<th>LPA</th>
<th>GBTF</th>
<th>Prop</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>44.09</td>
<td>44.33</td>
<td>44.91</td>
<td>44.67</td>
<td><strong>45.06</strong></td>
</tr>
<tr>
<td>2</td>
<td>41.45</td>
<td>41.62</td>
<td>42.31</td>
<td>42.45</td>
<td><strong>42.84</strong></td>
</tr>
<tr>
<td>3</td>
<td>43.17</td>
<td>43.28</td>
<td>43.91</td>
<td>43.82</td>
<td><strong>44.46</strong></td>
</tr>
<tr>
<td>4</td>
<td>37.70</td>
<td>37.30</td>
<td>38.49</td>
<td>38.34</td>
<td><strong>38.94</strong></td>
</tr>
<tr>
<td>5</td>
<td>44.21</td>
<td>44.14</td>
<td>44.51</td>
<td>44.70</td>
<td><strong>45.09</strong></td>
</tr>
<tr>
<td>6</td>
<td>41.04</td>
<td>41.17</td>
<td>41.61</td>
<td>41.97</td>
<td><strong>42.46</strong></td>
</tr>
<tr>
<td>7</td>
<td>44.74</td>
<td>44.80</td>
<td>44.91</td>
<td>45.45</td>
<td><strong>45.85</strong></td>
</tr>
<tr>
<td>8</td>
<td>41.66</td>
<td>42.01</td>
<td>42.49</td>
<td>42.79</td>
<td><strong>43.20</strong></td>
</tr>
<tr>
<td>9</td>
<td>42.42</td>
<td>42.56</td>
<td>42.79</td>
<td>43.18</td>
<td><strong>43.54</strong></td>
</tr>
<tr>
<td>10</td>
<td>40.58</td>
<td>40.95</td>
<td>40.99</td>
<td>41.18</td>
<td><strong>41.78</strong></td>
</tr>
<tr>
<td>11</td>
<td>39.65</td>
<td>39.84</td>
<td>39.98</td>
<td>40.30</td>
<td><strong>40.55</strong></td>
</tr>
<tr>
<td>12</td>
<td>37.30</td>
<td>37.76</td>
<td>37.67</td>
<td>37.87</td>
<td><strong>38.02</strong></td>
</tr>
<tr>
<td>Avg</td>
<td>41.50</td>
<td>41.61</td>
<td>42.05</td>
<td>42.23</td>
<td><strong>42.65</strong></td>
</tr>
</tbody>
</table>

Variance of Color Differences (VCD) [3]. The proposed algorithm has the best CPSNR for every image in the test set. It outperforms the closest method (GBTF) by 0.42 dB on average. The comparison results are summarized in Table 1 and a sample image region is shown in Figure 2.

Fig. 2: Fence region from image no. 8 (lighthouse).

4. CONCLUSION

In this paper, we demonstrated that the relationship between color gradients at different scales can be used to develop a high quality CFA interpolation method that is easy to implement. Experimental results show that the proposed method outperforms other available algorithms by a clear margin in terms of CPSNR. Further research efforts will focus on improving the results and applying the multiscale gradients idea to other image processing problems.

5. REFERENCES


