A RECONSTRUCTION ALGORITHM FOR MULTI-SPECTRAL IMAGE DEMOSAICING

Hemant Kumar Aggarwal Angshul Majumdar
Indraprastha Institute of Information Technology-Delhi Delhi, India
email: {hemanta, angshul}@iiitd.ac.in

Rabab Ward
Electrical and Computer Engineering Department
University of British Columbia, Canada
email: rababw@ece.ubc.ca

ABSTRACT
We have proposed a method by which compact and low-cost multi-spectral cameras can be designed based on the concept of single-sensor cameras. A filter pattern have been proposed to capture intensity values from multiple bands by using single-sensor. Our approach is based on representing central pixel in a window as a linear combination of neighboring intensity values from same and other bands. The proposed method can be applied to multi-spectral images with few bands. We did comparison of our method with two existing algorithms using peak signal to noise ratio (PSNR) and structural-similarity (SSIM) and found that experimental results were better both visually as well as quantitatively.

KEY WORDS
Multi-spectral Images, Demosaicing, Least-Square, PSNR, Structural-Similarity.

1 Introduction
Multi-spectral images are those images which have intensity values from multiple bands of electromagnetic spectrum ranging from ultraviolet to thermal band. A multi-spectral image have more information about an object than normal RGB image because it captures not only the wavelengths of visible band but also many other wavelengths of electromagnetic spectrum. There are several applications of multispectral images in various domains such as in military, remote sensing [1] and medical imaging [2].

The sensors used for multi-spectral imaging are optical-mechanical devices such as multispectral scanner used in Landsat satellite. Due to optical and mechanical parts in these sensors the size and cost of multi-spectral cameras are very high as compare to digital cameras which are based on silicon chips. The cost and size of these cameras can be reduced if they can capture light by using design techniques similar to digital cameras for RGB color imaging.

Many single-sensor digital cameras capture the light using Bayer color-filter-array (CFA) [3] which is based on the property that human-eye is more sensitive to green color than red and blue color. There are various demosaicing algorithms [4, 5] to reconstruct full color image from the raw image captured using Bayer pattern. Raw images captured using Bayer pattern have 50% samples from green band and 25% each of red and blue band but in case of multi-spectral imaging we do not know which band should be sampled more. Therefore we have chosen to give equal importance to all bands and sampling has been done uniformly for each band. Our proposed filter pattern for sensors of multispectral cameras will take samples uniformly from consecutive bands one-by-one. Figure 1a shows an example filter which can be applied on image-sensor to capture samples from three different consecutive bands of multi-spectral image. In this example each pixel have intensity value only from one band and other two band values need to be interpolated for complete reconstruction. This example filter pattern has been shown to capture red, green and blue intensity values but these can be any consecutive bands of multi-spectral image.

A Binary-tree based multi-spectral filter pattern has been proposed in [6] where authors have explored about the edge correlation to find missing intensity values in other bands. Spectral channel differences and bilinear interpolation based algorithm have been proposed in [7] where authors have also proposed $3 \times 2$ filter array for capturing multi-spectral images using CCD sensors. The paper [8] do a comparative study of some of the algorithms proposed for demosaicing of Bayer-pattern based images. Many of these approaches are based on properties and assumptions about the red, green and blue bands.

2 Proposed Method
Gradient-corrected bilinear interpolation technique [4] finds the filters which can be used to interpolate missing intensity values of other bands. In that technique once the filters have been found then they can be applied on each pixel in linear time to interpolate missing band values. Similarly our proposed technique can find the filters which can be applied to interpolate missing band values at each pixel in linear time.

Considering the example of a three band filter pattern shown in Figure 1a, we can see that there are three different repeating patterns as shown in Figures 1b, 1c and 1d. For each of the three patterns we can explore the neighborhood relationship to estimate the other two band values. It means that we need to find contribution of each of the nine available neighboring intensity values for the estimation of
unavailable values of other bands.

We describe our technique with the $3 \times 3$ neighborhood of a pixel and represent the intensity values of central pixel as a linear combination of neighboring intensity values. For example pattern of Figure 1b do not have two intensity values corresponding to red and green band at the central pixel. These two values can be represented as linear combination of neighboring values as shown in the equations below:

$$r_{22} = u_1^T \alpha$$

(1)

$$g_{22} = u_2^T \beta$$

(2)

$$u_1 = [r_{11} \ g_{12} \ b_{13} \ g_{21} \ b_{22} \ r_{23} \ b_{31} \ r_{32} \ g_{33}]^T$$

$$\alpha^T = [\alpha_{11} \ \alpha_{12} \ \alpha_{13} \ \alpha_{21} \ \alpha_{22} \ \alpha_{23} \ \alpha_{31} \ \alpha_{32} \ \alpha_{33}]$$

$$\beta^T = [\beta_{11} \ \beta_{12} \ \beta_{13} \ \beta_{21} \ \beta_{22} \ \beta_{23} \ \beta_{31} \ \beta_{32} \ \beta_{33}]$$

Here $r_{ij}$, $g_{ij}$, $b_{ij}$ represents the intensity of red, green, and blue bands respectively at location $(i,j)$. $\alpha$ and $\beta$ are the vectors which determine the contribution of each neighboring value for the estimation of intensity of red $r_{22}$ and green $g_{22}$ band at the point $(2,2)$. $u_1$ is the vector representing neighborhood intensity values. Each window with filter pattern 1b can be written in this manner and we will get overdetermined system of linear equations:

$$y_1 = A^T \alpha$$

(3)

$$y_2 = A^T \beta$$

(4)

$$A = [u_1 \ u_2 \ \ldots \ u_n]$$

Here $y_1$ and $y_2$ are the vectors which have intensity values from the original full color images. Once we have vector $y_1$ and the tall matrix $A$, we can solve the overdetermined system of equations using least-square method to find the vectors $\alpha$ and $\beta$. These two vectors can be used in equations 1 and 2 to interpolate red and green band values. Similar procedure can be applied for patterns 1c and 1d to interpolate unknown intensity values. In this manner we get the fractions by which each neighboring pixel will contribute for the estimation of unavailable values. This procedure can be applied on multi-spectral images with few bands to obtain the filters to interpolate missing intensity values.

### 3 Experiments

We did all the experiments using matlab software and kodak image dataset [9]. The kodak image dataset have color images of visibal band only but the proposed method will work with multi-spectral dataset as well since we have not used any property of the visible band in our proposed technique. We randomly took six full color images from the dataset and applied the filter of Figure 1a to get simulated raw images.

The simulated raw image have many repetitions of filter patterns. Each of these $3 \times 3$ neighborhoods of raw image form rows of matrix $A$ in equations 3 and 4. The vector $y_1$ of equation 3 have values from original red band corresponding to central pixel of each window. Similarly the vector $y_2$ of equation 4 can be formed from the values of original green band. Knowing $y_1$, $y_2$ and matrix $A$ we solved overdetermined system of linear equations to get vectors $\alpha$ and $\beta$. The vectors $\alpha$ and $\beta$ were then used for the estimation of red and green band values in the pattern 1b. Therefore for one filter pattern we solved two systems of linear equations and for total three patterns we solved six systems of linear equations to get the resulting vectors representing neighborhood relationship. Table 1 shows these six resulting filters for each of the repeating pattern.

We applied proposed filters of Table 1 on twenty-three images of the Kodak images[9]. Firstly the images were simulated to raw format as would be captured by camera-sensor using our proposed filter pattern of Figure 1 and then filters of Table 1 were applied on each of the raw images to

![Figure 1. Proposed pattern to capture raw image and three different repeating patterns occurring in the image](image-url)
get full-color reconstructed images. We used peak signal to noise ratio (PSNR) and structural-similarity (ssim) [10] to quantitatively measure quality of reconstruction by our proposed method. Structural-similarity was calculated using implementation available here [11].

4 Results

Figure 2 shows the result of applying our proposed filters on the portion of ‘lighthouse’ image from the kodak image dataset [9]. Figure 2a is the portion of original full color image, Figure 2b is the simulated bayer pattern image, Figure 2c is the image in our proposed filter pattern. Figure 2d and 2e are the results of applying bilinear interpolation and the algorithm of [4] on the simulated bayer pattern image. Figure 2f shows the reconstructed image which we get after applying filters of Table 1 on our proposed filter pattern.

It is visually clear from Figure 2 that our technique produces better results and it does not have image artifacts such as moiré pattern. Unlike bi-linear interpolation, our interpolation technique takes into consideration not only same band values but also the neighboring band values therefore our results are better than some other techniques.

Table 2 shows quantitative comparison of our method with bilinear interpolation and algorithm of [4]. The first column is showing the image number which is referring to each of the 23 images of kodak dataset. The average PSNR value for our technique is 35.7427dB which is highest as compare to both, approach [4] and bilinear interpolation. The structural similarity for our approach is higher than bilinear interpolation and is less by 0.0093 as compare to approach [4].

5 Conclusions and Future Work

Our results shows that effective interpolation can be achieved by exploring neighborhood relationship of nearby bands. The technique can be very useful in reducing number of sensors required in multi-spectral imaging and consequently cost and size of the cameras can also be reduced. In our study we did not used any property of human visual system such as those in Bayer pattern and some demosaicing algorithms where green band is given more importance then other two. We have done our experiments with three band color images but the same technique will work for multi-spectral images with few bands as well because any property of specific for a particular band of electromagnetic spectrum have not been considered.

The reconstruction quality can be measured by visual inspection which is subjective but have more significance than quantitative measures. As clear from Figure 2 our reconstruction results are very good as compare to other two algorithms. However the average structural-similarity index for our approach is slightly less (0.0093dB) than that approach in [4] but our approach is expandable to multi-spectral images. Moreover structural-similarity index have been developed by considering human visual system so there is need to develop other evaluation metric for multi-spectral image reconstruction methods.

More experiments need to be done with multi-spectral image datasets with different window sizes, say with window of $5 \times 5$ and by capturing more than three band values using single-sensor. The performance of multi-spectral imaging can also be measured using subjective human
Table 2. PSNR and structural similarity calculations for kodak image dataset

<table>
<thead>
<tr>
<th>Image num</th>
<th>Peak Signal to Noise Ratio</th>
<th>Structural-Similarity Index</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>34.044</td>
<td>28.539</td>
</tr>
<tr>
<td>2</td>
<td>36.050</td>
<td>33.673</td>
</tr>
<tr>
<td>3</td>
<td>38.591</td>
<td>33.857</td>
</tr>
<tr>
<td>4</td>
<td>36.801</td>
<td>34.537</td>
</tr>
<tr>
<td>5</td>
<td>32.235</td>
<td>28.617</td>
</tr>
<tr>
<td>6</td>
<td>34.988</td>
<td>30.838</td>
</tr>
<tr>
<td>7</td>
<td>37.886</td>
<td>34.770</td>
</tr>
<tr>
<td>8</td>
<td>32.343</td>
<td>24.524</td>
</tr>
<tr>
<td>9</td>
<td>38.629</td>
<td>32.425</td>
</tr>
<tr>
<td>10</td>
<td>37.351</td>
<td>32.478</td>
</tr>
<tr>
<td>11</td>
<td>35.229</td>
<td>31.147</td>
</tr>
<tr>
<td>12</td>
<td>39.862</td>
<td>34.600</td>
</tr>
<tr>
<td>13</td>
<td>29.111</td>
<td>26.523</td>
</tr>
<tr>
<td>14</td>
<td>33.502</td>
<td>30.533</td>
</tr>
<tr>
<td>15</td>
<td>36.133</td>
<td>31.349</td>
</tr>
<tr>
<td>16</td>
<td>38.822</td>
<td>34.251</td>
</tr>
<tr>
<td>17</td>
<td>39.971</td>
<td>33.597</td>
</tr>
<tr>
<td>18</td>
<td>32.659</td>
<td>29.423</td>
</tr>
<tr>
<td>19</td>
<td>36.571</td>
<td>29.057</td>
</tr>
<tr>
<td>20</td>
<td>36.921</td>
<td>32.490</td>
</tr>
<tr>
<td>21</td>
<td>34.420</td>
<td>30.344</td>
</tr>
<tr>
<td>22</td>
<td>35.428</td>
<td>31.335</td>
</tr>
<tr>
<td>23</td>
<td>37.532</td>
<td>33.129</td>
</tr>
<tr>
<td>Average</td>
<td>35.742</td>
<td>31.393</td>
</tr>
</tbody>
</table>

based testing.

References


