Demosaicing of images obtained from single-chip imaging sensors in YUV color space

Pattern Recognition Letters
vol. 26, 2005
Jayanta Mukherjee, Manfred K. Lang, S.K> Mitra

School of Electrical Engineering and Computer Science
Kyungpook National Univ.
Abstract

◆ The presentation techniques
  – To obtain the full color image using interpolating the CFA data in the YUV color space
  – Using the concept of sub-band DCT computation
  – Using the median filtering of the chrominance components

◆ The advantages
  – The speed of computations, quality of the image reconstruction, storage requirement, and false color suppression
Introduction

- The color filter array image

Fig. 1. The Bayer pattern and CFA image.
The several advantage

- To directly obtain the DCT based JPEG compression scheme using the YUV color space
- The computation fast
- To get the chrominance components separated from the luminance one
  - To reduce false color
Color interpolation in the YUV Space

- From RGB to YUV

\[ y = Y(r, g, b) \]
\[ u = U(r, g, b) \]  \hspace{1cm} (1)
\[ v = V(r, g, b) \]

\[
\begin{bmatrix}
y \\
u \\
v
\end{bmatrix} =
\begin{bmatrix}
0.299 & 0.587 & 0.114 \\
-0.173 & -0.339 & 0.511 \\
0.511 & -0.428 & -0.083
\end{bmatrix}
\begin{bmatrix}
r \\
g \\
b
\end{bmatrix} \tag{2}
\]
A 2x2 block in the CFA

Fig. 2. A 2x2 block in the CFA.

Fig. 3. Pixels in the mosaiced image for the corresponding 2x2 block.
The proposed interpolation algorithm
– Algorithm Simple_YUV_interpolation (SYUV)

\[ g_{av} = \left( g_{11} + g_{22} \right) / 2 \]
\[ y = Y(r_{12}, g_{av}, b_{21}) \]
\[ u = U(r_{12}, g_{av}, b_{21}) \]
\[ v = V(r_{12}, g_{av}, b_{21}) \]
Fig. 4. Reconstructed images by SYUV: (a) Statue (original), (b) by SYUV, (c) Sail (original), (d) by SYUV, (e) Pepper (original), and (f) by SYUV
The performance of the algorithm

Table 1. Recover of individual components using the SYUV interpolation algorithm.

<table>
<thead>
<tr>
<th>Images</th>
<th>PSNR</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Y (dB)</td>
</tr>
<tr>
<td>Statue</td>
<td>28.49</td>
</tr>
<tr>
<td>Lighthouse</td>
<td>24.92</td>
</tr>
<tr>
<td>Window</td>
<td>27.71</td>
</tr>
<tr>
<td>Sail</td>
<td>27.67</td>
</tr>
<tr>
<td>Pepper</td>
<td>26.83</td>
</tr>
</tbody>
</table>
Computations through ‘green’ interpolation
– YUV_through_Green_Interpolation (YUVG)

\[
g_{av} = (g_{11} + g_{12} + g_{12} + g_{22}) / 4
\]
\[
y_{11} = Y(r_{12}, g_{11}, b_{21})
\]
\[
y_{12} = Y(r_{12}, g_{12}, b_{21})
\]
\[
y_{21} = Y(r_{12}, g_{21}, b_{21})
\]
\[
y_{22} = Y(r_{12}, g_{22}, b_{21})
\]
\[
u = U(r_{12}, g_{av}, b_{21})
\]
\[
v = V(r_{12}, g_{av}, b_{21})
\]
Green Interpolation

- Using the Hamilton and Adams method

\[
G_5 = \begin{cases} 
\frac{G_4 + G_6}{2} + \frac{-A_3 + 2A_5 - A_7}{2} & \text{if } \alpha < \beta \\
\frac{G_2 + G_8}{2} + \frac{-A_1 + 2A_5 - A_9}{2} & \text{if } \alpha > \beta \\
\frac{G_2 + G_4 + G_6 + G_8}{4} + \frac{-A_1 - A_3 + 4A_5 - A_7 - A_9}{2} & \text{if } \alpha = \beta
\end{cases}
\]

\[
\alpha = \text{abs}(-A_3 + 2A_5 - A_7) + \text{abs}(G_4 - G_6)
\]

\[
\beta = \text{abs}(-A_1 + 2A_5 - A_9) + \text{abs}(G_2 - G_8)
\]

Ref. Fig. Sample Bayer neighborhood,

\(A_j = \text{chrominance (blue/red)}, G_j = \text{luminance.}\)
Fig. 5. Reconstructed images by YUVG: (a) Statue, (b) Sail and (c) Pepper.
The performance of the algorithm

Table 2. Recover of individual components using the YUVG interpolation algorithm.

<table>
<thead>
<tr>
<th>Images</th>
<th>PSNR (Y dB)</th>
<th>PSNR (U dB)</th>
<th>PSNR (V dB)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Statue</td>
<td>28.49</td>
<td>32.85</td>
<td>32.13</td>
</tr>
<tr>
<td>Lighthouse</td>
<td>24.92</td>
<td>30.68</td>
<td>29.67</td>
</tr>
<tr>
<td>Window</td>
<td>27.71</td>
<td>32.50</td>
<td>31.25</td>
</tr>
<tr>
<td>Sail</td>
<td>27.67</td>
<td>33.16</td>
<td>32.14</td>
</tr>
<tr>
<td>Pepper</td>
<td>26.83</td>
<td>29.35</td>
<td>30.50</td>
</tr>
</tbody>
</table>
Post-processing U and V components

- The image in the YUV color space
  - To separate achromatic (Y) and chromatic components (U, V)
- The suppression of false colors
  - Using median filtering with 3x3 mask

Table 3. Recover of individual components using the YUVGM interpolation algorithm.

<table>
<thead>
<tr>
<th>Images</th>
<th>PSNR</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Y (dB)</td>
</tr>
<tr>
<td>Statue</td>
<td>32.91</td>
</tr>
<tr>
<td>Lighthouse</td>
<td>30.77</td>
</tr>
<tr>
<td>Window</td>
<td>32.53</td>
</tr>
<tr>
<td>Sail</td>
<td>33.25</td>
</tr>
<tr>
<td>Pepper</td>
<td>29.46</td>
</tr>
</tbody>
</table>


<table>
<thead>
<tr>
<th>U (dB)</th>
<th>V (dB)</th>
</tr>
</thead>
<tbody>
<tr>
<td>31.50</td>
<td>32.13</td>
</tr>
<tr>
<td>28.90</td>
<td>29.67</td>
</tr>
<tr>
<td>30.41</td>
<td>31.25</td>
</tr>
<tr>
<td>31.46</td>
<td>32.14</td>
</tr>
<tr>
<td>30.22</td>
<td>30.50</td>
</tr>
</tbody>
</table>
Fig. 6. Absolute error between original (sub-sampled) and reconstructed chrominance components YUVG interpolation algorithm.
(a) Error in the U component and (b) error in the V component.
Fig. 7. Reconstructed images by YUVGM: (a) Statue, (b) Sail and (c) Pepper.
Sub-band interpolation of U and V components

- To improve the interpolated image
  - Using up-sampling of the U and V components

Table 4. Recover of individual components using the YUVGMSB interpolation algorithm.
Fig. 8. Reconstructed images by YUVGMSB: (a) Statue, (b) Sail and (c) Pepper.
Computational overheads

– To implement on a Linux-7.3 Work station
– Pentium-III with 550 MHz clock

Table 5. Average pixel processing times (in μs).

<table>
<thead>
<tr>
<th></th>
<th>YUVG</th>
<th>YUVGM</th>
<th>YUVGMSB</th>
</tr>
</thead>
<tbody>
<tr>
<td>SYUV</td>
<td>1.75</td>
<td>4.37</td>
<td>11.99</td>
</tr>
<tr>
<td></td>
<td>1.07</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Performance evaluation of the algorithms

- To compare the performances of the proposed interpolation algorithms
  - Quality of the image reconstructions
  - Memory requirement
  - Speed of computations
  - False color suppression
Quality of the image reconstructions

– CPSNR (Composite-Peak-Signal-to-Noise-Ratio)

\[
\text{CPSNR} = 20 \log_{10} \left( \frac{255}{\sqrt{\sum_{s=1}^{3} \sum_{x} \sum_{y} \left( I_s(x,y) - I'_s(x,y) \right)^2 / 3MN}} \right)
\]

where \( I_s(x,y), \ s = 1, 2, 3 \) : The spectral component of a benchmark image

\( I'_s(x,y), \ s = 1, 2, 3 \) : The respective reconstructed spectral component
– PEINR (Peak-Edge-Intensity-to-Noise-Ratio)

\[
\text{PEINR} = 20 \times \log_{10} \left( \frac{255}{\sqrt{\sum_{x} \sum_{y} e(x, y) \times (I_s(x, y) - I'_s(x, y))^2}} \right)
\]

where \( I_s(x, y), \ s = 1, 2, 3 \) : The spectral component of a benchmark image
\( I'_s(x, y), \ s = 1, 2, 3 \) : The respective reconstructed spectral component
\( e(x, y) \) : An edge map
Fig. 9. Reconstructed images by bilinear interpolation (BI) and Kimmel’s algorithm (EDCRAC). (a) Statue (BI), (b) Statue (EDCRAC), (c) Sail (BI), (d) Sail (EDCRAC), (e) Pepper (BI), and (f) Pepper (EDCRAC).
Table 6. CPSNR for different interpolation techniques.

<table>
<thead>
<tr>
<th>Images</th>
<th>YUVGM (dB)</th>
<th>YUVGMSB (dB)</th>
<th>Bilinear (dB)</th>
<th>EDCRAC (dB)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Statue</td>
<td>31.87</td>
<td>31.97</td>
<td>27.85</td>
<td>31.58</td>
</tr>
<tr>
<td>Lighthouse</td>
<td>28.85</td>
<td>29.10</td>
<td>25.09</td>
<td>30.40</td>
</tr>
<tr>
<td>Window</td>
<td>30.16</td>
<td>30.55</td>
<td>26.84</td>
<td>32.69</td>
</tr>
<tr>
<td>Sail</td>
<td>31.82</td>
<td>32.02</td>
<td>27.69</td>
<td>33.45</td>
</tr>
<tr>
<td>Pepper</td>
<td>26.52</td>
<td>26.80</td>
<td>25.45</td>
<td>25.04</td>
</tr>
</tbody>
</table>

Table 7. PEINR for different interpolation techniques.

<table>
<thead>
<tr>
<th>Images</th>
<th>YUVGM (dB)</th>
<th>YUVGMSB (dB)</th>
<th>Bilinear (dB)</th>
<th>EDCRAC (dB)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Statue</td>
<td>17.95</td>
<td>17.92</td>
<td>12.91</td>
<td>13.57</td>
</tr>
<tr>
<td>Lighthouse</td>
<td>18.80</td>
<td>18.59</td>
<td>19.61</td>
<td>17.35</td>
</tr>
<tr>
<td>Window</td>
<td>16.44</td>
<td>16.26</td>
<td>15.17</td>
<td>24.18</td>
</tr>
<tr>
<td>Sail</td>
<td>18.79</td>
<td>18.96</td>
<td>18.46</td>
<td>20.97</td>
</tr>
<tr>
<td>Pepper</td>
<td>14.43</td>
<td>14.40</td>
<td>11.51</td>
<td>11.77</td>
</tr>
</tbody>
</table>
Computation speed

Table 8. Average pixel processing times.

<table>
<thead>
<tr>
<th>YUVGM</th>
<th>YUVGMSB</th>
<th>Bilinear</th>
<th>EDCRAC</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.37</td>
<td>11.99</td>
<td>2.97</td>
<td>9.65</td>
</tr>
</tbody>
</table>

Memory requirement

- The conventional algorithm: \( 4 \times (M \times N) \)

- Both YUVGM and YUVGMSB: \( 2 \times (M \times N) + 2 \times \left( \frac{M}{2} \times \frac{N}{2} \right) \)

where \( M \times N \): A mosaiced pattern of size
False color suppression

Fig. 10. Example of false color suppressions in an enlarged part of Statue.
(a) YUVGM, (b) YUVGMSB, (c) BI, and (d) EDCRAC
Fig. 11. Example of false color suppressions in an enlarged part of Sail.
(a) YUVGM,
(b) YUVGMSB,
(c) BI, and
(d) EDCRAC
Fig. 12. Example of false color suppressions in an enlarged part of Pepper.
(a) YUVGM,
(b) YUVGMSB,
(c) BI, and
(d) EDCRAC
Conclusions

◆ The proposed algorithm
  – To obtain the full color image using interpolating the CFA data in the YUV color space

◆ The advantages
  – The speed of computations, quality of the image reconstruction, storage requirement, and false color suppression
The bilinear interpolation

\[ R_b = \frac{R_1 + R_3 + R_4 + R_5}{4} \]

\[ R_g = \frac{R_1 + R_2}{2} \]

\[ B_b = \frac{B_1 + B_2}{2} \]

\[ B_g = \frac{B_1 + B_2 + B_3 + B_4}{4} \]

\[ G_b = \frac{G_1 + G_2 + G_3 + G_4}{4} \]

\[ G_g = \frac{G_1 + G_2 + G_3 + G_4}{4} \]
◆ The interpolation equation of Kimmel

– The green color at the center pixel

\[
G_{ij} = \frac{w_{ij}^{i-1,j} G_{i-1,j} + w_{ij}^{i+1,j} G_{i+1,j} + w_{ij}^{i,j-1} G_{i,j-1} + w_{ij}^{i,j+1} G_{i,j+1}}{w_{ij}^{i-1,j} + w_{ij}^{i+1,j} + w_{ij}^{i,j-1} + w_{ij}^{i,j+1}}
\]

– The weighting factors

\[
w_{ij}^{i+1,j} = 1/(1 + Dx_{i,j}^2 + Dx_{i+1,j}^2)^{1/2}, \quad Dx_{i,j} = \frac{P_{i+1,j} - P_{i-1,j}}{2}, \quad Dy_{i,j} = \frac{P_{i,j+1} - P_{i,j-1}}{2}
\]

where \( w \); the weight factors,

\( P \); green, blue, or red pixels in the kernel

\( Dx_{i,j} \) and \( Dy_{i,j} \); the directional derivatives in horizontal and vertical direction
CFA interpolation

Fig. 6. (a) CFA interpolation kernel; (b) red center pixel, (c) blue center pixel.

\[
D_x G_{i,j} = \frac{G_{i+1,j} - G_{i-1,j}}{2\Delta x}
\]

\[
D_y G_{i,j} = \frac{G_{i,j+1} - G_{i,j-1}}{2\Delta y}
\]
– The Kimmel’s chrominance interpolation equation

  • The blue interpolation at red locations

\[
B_{i,j} = G_{i,j} \frac{w_{i+1,j+1} \frac{B_{i+1,j+1}}{G_{i+1,j+1}} + w_{i-1,j-1} \frac{B_{i-1,j-1}}{G_{i-1,j-1}} + w_{i,j+1} \frac{B_{i-1,j+1}}{G_{i-1,j+1}} + w_{i,j-1} \frac{B_{i+1,j-1}}{G_{i+1,j-1}}}{w_{i+1,j+1} + w_{i-1,j-1} + w_{i,j+1} + w_{i,j-1}}
\]

  • The blue interpolation at green locations

\[
B_{i,j} = G_{i,j} \frac{w_{i+1,j} \frac{B_{i+1,j}}{G_{i+1,j}} + w_{i-1,j} \frac{B_{i-1,j}}{G_{i-1,j}} + w_{i,j+1} \frac{B_{i-1,j+1}}{G_{i-1,j+1}} + w_{i,j-1} \frac{B_{i+1,j-1}}{G_{i+1,j-1}}}{w_{i+1,j} + w_{i-1,j} + w_{i,j+1} + w_{i,j-1}}
\]