Image Interpolation Based On Multi Scale Gradients

Sreegadha G.S*

 Dept.of computer science,TKM Institute of Technology,Karuvil
 Kollam-691601,Kerala,India.

Abstract

Most low-cost consumer grade digital camera systems are currently designed to sense only one color component per image pixel and interpolate the other missing color components (at each pixel) during reconstruction. The sensing process, which employs a Color Filter Array (CFA), maps each pixel to a single color based on a color pattern. The CFA color pattern and the interpolation process (widely known as demosaicing) have a significant impact on the quality of the reconstructed image. In this paper, a directional CFA interpolation method that is based on multiscale color gradients. The method is easy to implement since it does not make any hard decision, noniterative and threshold free. This method, the horizontal and vertical color difference estimates are blended based on the ratio of the total absolute values of vertical and horizontal color difference gradients over a local window. Method is applied to Bayer and Lukac pattern with great results which shows that the relationship between gradients at different scales can be a very effective feature to optimally combine directional estimates. Result can be used to compare efficiency of both pattern in this method.

Keywords: ColorFilterArray(CFA) interpolation,demosaicing,directional interpolation, multiscale color gradients.

1. Introduction

Digital cameras have become more and more popular in consumer electronics market. In order to economize the hardware cost, instead of using three sensors, most digital cameras capture a color image with a signal sensor imaging pipeline based on the well-known Bayer CFA, where each pixel in the captured image has only one measured. In order to recover the full color image from the input mosaic image, the demosaicing process is used to
estimate the other two color channels for each pixel. Demosaicing is an important part of the image processing pipeline in digital cameras. The failure of the employed demosaicing algorithm can degrade the overall image quality considerably. The CFA pattern layout plays an important role in the design of a CFA interpolation algorithm. Many different CFA patterns have been proposed over the years. While some of these are about a conference proceedings comprised of pure RGB channels, like Bayer and Lukac patterns shown in Fig. 1 [1], [4], others feature linear combinations of RGB channels [3], [4], [14]. The aim of a demosaicing algorithm is to reconstruct a full color image (i.e. a full set of color triples) from the spatially under sampled colour channel output from the CFA. The algorithm should have different advantages: such as it’s avoidance of the introduction of false color artifacts, such as chromatic aliases, zipperng (abrupt unnatural changes of intensity over a number of neighboring pixels) and purple fringing, provide maximum preservation of the image resolution, low computational complexity for fast processing or efficient in-camera hardware implementation, amenability to analysis for accurate noise reduction.

![Fig. 1. Bayer and Lukac mosaic patterns.](image)

The simplest way to address the demosaicing problem would be to treat each color channel separately and interpolate missing samples using a spatially invariant method such as bilinear or bicubic interpolation. However, such a solution would lead to false color artifacts wherever there is a sudden color change. The quality can be improved by applying the interpolation over color differences to take advantage of the correlation between the color channels[14]. However, the lack of spatial adaptiveness[13] would still limit the interpolation performance. So my proposed prediction algorithm provides a solution to the mentioned problem. The main features of this algorithm is that it’s applied to Bayer and Lukac pattern with great results which shows that the relationship between gradients at different scales can be a very effective feature to optimally combine directional estimates.

2. Related work

Demosaicing is an important part of the image processing in digital cameras. The failure of the employed demosaicing algorithm can degrade the overall image quality. So it has been an active research area for many years. Although there has been recent efforts to introduce generalized demosaicing algorithms, most demosaicing solutions in the literature are developed for the Bayer pattern[1]. The quality can be improved by applying the interpolation over color differences to take advantage of the correlation between the color channels[1]. However, the lack of spatial adaptiveness would still limit the interpolation performance.

R. Lukac and K.N. Plataniotis [2] used a normalized color-ratio model suitable for color filter array (CFA) interpolation. The first solution utilizes linear shifts to alleviate effects of edge variations in the interpolator’s input. The second solution take advantages of both the linear scaling and shifting operations to normalize the color-ratio variations in the interpolator’s input. Xin Li[3] used a fast and high-performance iterative algorithm for color filter array (CFA) demosaicing. The major contributions of this work include a new iterative demosaicing algorithm in the color difference domain and second one is a spatially adaptive stopping criterion for suppressing color artifacts.
misregistration and zipper artifacts in the demosaiced images. Ibrahim Pekucuksen, Yucel Altunbasak[4][13] used a simple edge strength filter to interpolate the missing color values. The solution outperforms other available algorithms for the Lukac pattern in terms of both objective and subjective comparison.

Daniele Menon, Stefano Andriani, and Giancarlo Calvagno[5] used novel approach to demosaicing based on directional filtering and a posteriori decision. A refining step is included to further improve the resulting reconstructed image. Wemmiao Lu and Yap-Peng Tan[6] introduced a new CFA demosaicking method that consists of two successive steps: an interpolation step that fills in missing color values in a progressive fashion by exploiting the spectral and spatial correlations among neighboring pixels, and a post-processing step that incorporates spectral correlation with median filtering of inter-channel differences to suppress demosaicking artifacts. C. Naga raju, K. Subba reddy and C. Sunee a[7] proposes a method of CFA interpolation that combines information from the green image with the subsampled red and blue images to attack these problems. Kuo-Liang Chung, Wei-jen Yang, Wen-Ming Yan and Chung-chou Wang[8] proposed a method which does not use demosaicing processing, this first propose a new approach to extract more accurate gradient/edge information on mosaic images directly. Next, based on spectral–spatial correlation. Henrique S. Malvar, Li-wei He, and Ross Cutler[9] presented a new demosaicing method for color interpolation of images captured from a single CCD using a Bayer color filter array. Nai-Xiang Lian, Lanlan Chang, Yap-Peng Tan and Vitali Zagorodnov[10] introduced a new approach which discussed two important observations for preserving high-frequency information in CFA demosaicking.

All the current approaches deal with demosaicing based on Bayer pattern. In the proposed system demosaicing based on Bayer and other RGB pattern (eg. Lukac pattern) and compare the performance in both patterns. Efficiency of each one to be find out. Here a directional CFA interpolation method that is based on multiscale color gradients[1]. The developed method is applied to Bayer and Lukac patterns with great results which shows that the relationship between gradients at different scales can be a very effective feature to optimally combine directional estimates. This method the horizontal and vertical color difference estimates are blended based on the ratio of the total absolute values of vertical and horizontal color difference gradients over a local window.

3. Proposed Algorithm

A. Algorithm Background

Gradients are useful for extracting directional data from digital images. In this method, the horizontal and vertical color difference estimates are blended based on the ratio of the total absolute values of vertical and horizontal color difference gradients over a local window.

The first step of the algorithm is to get initial directional color channel estimates. The Bayer pattern is comprised of blue&green and red&green rows and columns as depicted in Fig. 1. For red&green rows and columns in the input mosaic image, the directional estimates for the missing red and green pixel values are:

\[
\begin{align*}
G^H(i, j) &= \frac{G(i, j - 1) + G(i, j + 1)}{2} + \frac{2.2R(i, j) - R(i, j - 2) - R(i, j + 2)}{4} \\
R^H(i, j) &= \frac{G(i, j - 1) + G(i, j + 1)}{2} + \frac{2.2G(i, j) - G(i, j - 2) - G(i, j + 2)}{4} \\
G^V(i, j) &= \frac{G(i - 1, j) + G(i + 1, j)}{2} + \frac{2.2R(i, j) - R(i - 2, j) - R(i + 2, j)}{4} \\
R^V(i, j) &= \frac{G(i - 1, j) + G(i + 1, j)}{2} + \frac{2.2G(i, j) - G(i - 2, j) - G(i + 2, j)}{4},
\end{align*}
\]

where \(H\) and \(V\) denote horizontal and vertical directions and \((i, j)\) is the pixel location. For every pixel coordinate, now have a true color channel value and two directional estimates. By taking their difference, get the directional
The absolute color difference gradients at pixel coordinates \((i, j)\) are given by:

\[
\Delta_{g,r}^H(i, j) = \begin{cases} 
\tilde{G}^H(i, j) - R(i, j), & \text{if } G \text{ is interpolated} \\
G(i, j) - \tilde{R}^H(i, j), & \text{if } R \text{ is interpolated}
\end{cases}
\]

\[
\Delta_{g,r}^V(i, j) = \begin{cases} 
\tilde{G}^V(i, j) - R(i, j), & \text{if } G \text{ is interpolated} \\
G(i, j) - \tilde{R}^V(i, j), & \text{if } R \text{ is interpolated}
\end{cases}
\]

The color difference gradients calculated above are used to find weights for each direction. The horizontal color difference gradient equation above can be written in terms of red and green pixel values as follows:

\[
D^H(i, j) = |(G(i, j - 1) - \tilde{R}^H(i, j - 1)) - (G(i, j + 1) - \tilde{R}^H(i, j + 1))|
\]

\[
= \left| \frac{2G(i, j - 1) + G(i, j - 3) + G(i, j + 1)}{4} - \frac{R(i, j - 2) + R(i, j)}{2} \right|
\]

\[
- \left( \frac{2G(i, j + 1) + G(i, j - 1) + G(i, j + 3)}{4} - \frac{R(i, j) + R(i, j + 2)}{2} \right|
\]

Observe that there are \(R(i, j)\) terms present and they cancel out each other. Rearranging with respect to different color channels leaves with:

\[
D^H(i, j) = \left| \frac{R(i, j + 2) - R(i, j - 2)}{2} \right|
\]

\[
- \left( \frac{(G(i, j + 3) + G(i, j + 1)) - (G(i, j - 3) + G(i, j - 1))}{4} \right|
\]

There are two important observations that made on the equation above. First, color difference gradient corresponds to taking the difference between the available color channel values two pixels away from the target pixel, doing the same operation in terms of the other color channel by using simple averaging, and then finding the difference between these two operations as illustrated in the top portion of Figure 3. If these two color channels are changing in parallel with each other along this direction, then the resulting absolute value would be small. On the other hand, if there is an abrupt color change, then the result would be large and the color difference estimate along this direction would be given a small weight in combined color difference calculation. Second and more important observation is
that, these same operations can do at half the scale:

\[
D_h(i, j) = \left| \frac{G(i, j + 1) - G(i, j - 1)}{2} \right| - \left| \frac{(R(i, j + 2) + R(i, j) - (R(i, j - 2) + R(i, j))}{4} \right|,
\]

where \(D_h(i, j)\) denotes the horizontal estimation at half the scale. A smaller scale is more desirable because it allows the local color dynamics to be captured at a better resolution. Note that the available color channels are replaced at this scale, but still performing the same operations: Take the difference between the available color channel values one pixel (instead of two pixels) away from the target pixel. Do the same operation in terms of the other channel by using its closest samples, and then take the difference between these two. At this scale, the \(R(i, j)\) terms cancel each other out and left with:

\[
D_h(i, j) = \left| \frac{G(i, j + 1) - G(i, j - 1)}{2} \right| - \left| \frac{R(i, j + 2) - R(i, j - 2)}{4} \right|.
\]

Fig. 2. Relationship between the color difference gradients equation and the multiscale gradients equation

Observe that the first part of this equation is the green channel gradient, and the second part is the red channel gradient at twice the scale normalized by the distance between their operands as shown in the bottom part of Figure 3. Like the color difference gradient equation (Equation no. 5), this equation checks whether different color channel pixels along this direction are changing in agreement with each other or not. However, Expected this new equation to be more successful with combining the directional estimates because capture the color dynamics at a more local level and do it without resorting to any simple averaging. The fact that this equation combines two different scales of gradients together gave the idea that it should be possible to incorporate even more scales into the equation. However, since the locality will get weaker with each additional scale, the larger scales should contribute less to the result. The easiest way of doing that is to optimize the normalizing terms in the denominators. The final multiscale gradients equations for red & green rows and columns can be given as follows:
where the \( N_i \) terms are the normalizers. The equations are similar for blue & green rows and columns. Although Bayer mosaic pattern is a special case, this algorithm can be applied to other mosaic patterns with some modifications. This paper, the Lukac mosaic pattern proposed in [1]. An inspection on the Lukac mosaic pattern reveals that it is possible to take gradients in three directions as opposed to four on the Bayer pattern. While still having the horizontal component, the vertical one is gone and the diagonal components lean more towards the vertical direction. Based on this observation, some changes in equations needed to apply the Lukac patterns and get great results which shows that the relationship between gradients at different scales can be a very effective feature to optimally combine directional estimates.

B. Initial Green Channel Interpolation

Algorithm starts with interpolating the green channel. After updating the initial green channel interpolation results in one pass, the red and blue channels are filled in using the constant color difference assumption. The ratio between the vertical and horizontal multiscale gradients results over a local window is employed at every stage. For initial green channel interpolation, directional color difference estimates around every green pixel to be interpolated as given in Equation (2) and combine them adaptively:

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

\[
w_C = w_V + w_H
\]

\[
f = [1/4 \ 2/4 \ 1/4].
\]

For a local window size of 5 by 5, the weight for each direction is calculated as follows:

\[
w_V = 1 / \left( \sum_{k=1-i-2}^{i+2} \sum_{l=1-j-2}^{j+2} H(k, l) \right)^2
\]

\[
w_H = 1 / \left( \sum_{k=1-i-2}^{i+2} \sum_{l=1-j-2}^{j+2} H(k, l) \right)^2
\]

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).

C. Green Channel Update

After the directional color difference estimates are combined as explained in the previous section, we can directly calculate the green channel and move onto completing the other channels. However, it is possible to improve the green channel results by updating the initial color difference estimates. Consider the closest four neighbors to the target pixel with each one having its own weight

\[
\tilde{\Delta}_{g,r}(i, j) = \frac{1}{w_T} \left[ w_N \tilde{\Delta}_{g,r}(i, j) + w_S \tilde{\Delta}_{g,r}(i, j - 1) + w_E \tilde{\Delta}_{g,r}(i - 1, j) + w_W \tilde{\Delta}_{g,r}(i - 1, j - 1) \right]
\]

\[
w_T = w_N + w_S + w_E + w_W.
\]

(11)

Again, the weights ($w_N, w_S, w_E, w_W$) are calculated by finding the total multiscale color gradients over a local window. For a 3 by 5 window for horizontal and a 5 by 3 window for vertical components, the weight calculations can be given as follows

\[
w_N = 1/\left( \sum_{i=1}^{3} \sum_{j=1}^{5} D^{\sigma}(k, l) \right)^2
\]

\[
w_S = 1/\left( \sum_{k=1}^{5} \sum_{l=1}^{3} D^{\sigma}(k, l) \right)^2
\]

\[
w_W = 1/\left( \sum_{i=1}^{3} \sum_{j=1}^{5} D^{\theta}(k, l) \right)^2
\]

\[
w_E = 1/\left( \sum_{i=1}^{3} \sum_{j=1}^{5} D^{\theta}(k, l) \right)^2.
\]

(12)

Once the color difference estimate is finalized, add it to the available target pixel to obtain the estimated green channel value:

\[
\tilde{G}(i, j) = R(i, j) + \tilde{\Delta}_{g,r}(i, j)
\]

\[
\tilde{G}(i, j) = B(i, j) + \tilde{\Delta}_{g,b}(i, j).
\]

(13)

D. Red and Blue Channel Interpolation

For red and blue channel interpolation, first complete the missing diagonal samples i.e. red pixel values at blue locations and blue pixel values at red locations. These pixels are interpolated using the 7 by 7 filter.
where $\otimes$ denotes element-wise matrix multiplication and subsequent summation. The red and blue pixels at green locations are interpolated adaptively. In order to avoid repetitive weight calculations, we reuse the directional weights $(w_H, w_V)$ defined in Equation (10). 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:

$$
\begin{align*}
\tilde{R}_{i,j} &= G(i,j) \\
&\quad - \frac{w_V. (\tilde{G}(i-1,j) - R(i-1,j) + \tilde{G}(i+1,j) - R(i+1,j))}{2.(w_V + w_H)} \\
&\quad - \frac{w_H. (\tilde{G}(i,j-1) - \tilde{R}(i,j-1) + \tilde{G}(i,j+1) - \tilde{R}(i,j+1))}{2.(w_V + w_H)} \\
\tilde{B}(i,j) &= G(i,j) \\
&\quad - \frac{w_V. (\tilde{G}(i-1,j) - \tilde{B}(i-1,j) + \tilde{G}(i+1,j) - \tilde{B}(i+1,j))}{2.(w_V + w_H)} \\
&\quad - \frac{w_H. (\tilde{G}(i,j-1) - B(i,j-1) + \tilde{G}(i,j+1) - B(i,j+1))}{2.(w_V + w_H)}.
\end{align*}
$$

(15)

The equations for the blue vertical neighbor case are similar. With the completion of red and blue pixel values at green coordinates, obtain the full color image.

E. Application to the Lukac Pattern

Although designed for the Bayer mosaic pattern, the proposed method can be modified to be applied to other mosaic patterns. However, such an application may not be feasible for all mosaic patterns because of the restrictions dictated by the directional nature of approach. When the modification is feasible, an important question would be whether the changes needed to comply with the new pattern layout lead to a significant performance loss or not. To find out if one can outperform other available solutions on a different pattern layout, so modified the proposed algorithm for the Lukac pattern. Lukac mosaic pattern is similar to Bayer pattern in the sense that it consists of pure RGB components. When shift every other row in a Bayer pattern by one pixel to either side, obtain the Lukac pattern. Hence, the horizontal relationship between the pixels is still the same, but the vertical arrangement is significantly altered. As a result of this, it is not possible to take immediate vertical gradients anymore. However, observe that take vertical gradients when double the scale. So modified vertical multiscale gradients equation accordingly:
where the $M_i$ terms are the normalizers. The layout of the Lukac pattern also necessitates a change in vertical color difference estimation. Since all the required channel values are not available in the same column, estimate the missing values by taking simple average using samples from adjacent columns, shown at the bottom of the previous page. Another problem faced with the Lukac pattern was the mismatch between vertical and horizontal color difference estimates at green channel coordinates. Namely, the calculated vertical and horizontal color differences at these locations belong to different color pairs. That is why bring the needed vertical color difference estimate from the closest available resource:

\[
\begin{align*}
\hat{R}_V(i, j) &= \frac{R(i - 1, j - 1) + R(i - 1, j + 1) + R(i + 1, j)}{2} + \frac{G(i, j) - G(i - 2, j - 1) + G(i - 2, j + 1) - G(i + 2, j - 1) + G(i + 2, j + 1)}{2} \\
\hat{G}_V(i - 1, j - 1) &= \frac{G(i - 2, j - 1) + G(i, j - 2) + G(i, j)}{2} + \frac{R(i - 1, j - 1) - R(i - 3, j - 2) + R(i - 3, j) - R(i + 1, j - 2) + R(i + 1, j)}{2} \\
\hat{\Delta}_{g,b}(i, j) &= G(i - 1, j) - \hat{B}_V(i - 1, j) \\
\hat{\Delta}_{g,r}(i - 1, j) &= G(i, j) - \hat{R}_V(i, j).
\end{align*}
\]

(17)

Also, the combined color difference estimate equations are modified to bring the neighboring vertical estimates from two pixels away instead of one:

\[
\hat{\Delta}_{g,r}(i - 1, j) = [w_V f_v \hat{\Delta}_{g,r}^V(i - 3 : i + 1, j) + w_H \hat{\Delta}_{g,r}^H(i - 1, j - 2 : j)f_h]/w_C
\]

\[
w_C = w_V + w_H
\]

\[
f_h = [1/4 2/4 1/4]
\]

\[
f_v = [1/4 0 2/4 0 1/4].
\]

(18)
And finally, the red and blue channel interpolation requires modification as well. Estimate the missing red and blue samples using the closest color difference estimates. For the red channel interpolation, the pixels on green&blue rows use estimates from three neighbors and the ones on green&red rows use four:

\[
\hat{\Delta}_{g,r}(i, j) = \hat{\Delta}_{g,r}(i, j). (1 - w) \\
+ [w_N. (\hat{\Delta}_{g,r}(i - 2, j - 1) + \hat{\Delta}_{g,r}(i - 2, j + 1) \\
+ \hat{\Delta}_{g,r}(i - 4, j)]. / 3 w_S. (\hat{\Delta}_{g,r}(i + 2, j - 1) \\
+ \hat{\Delta}_{g,r}(i + 2, j + 1) + \hat{\Delta}_{g,r}(i + 4, j)] / 3 \\
w_E. \hat{\Delta}_{g,r}(i, j - 2) + w_W. \hat{\Delta}_{g,r}(i, j + 2)]. w / w_T \\
w_T = w_N + w_S + w_E + w_W. 
\]

\[\text{(19)}\]

Although needed to make several changes to apply the algorithm to the Lukac pattern, the main structure of the method is maintained.

4. Experimental Results

In the proposed system demosaicing based on Bayer and Lukac RGB pattern and compare the performance in both patterns. Here A directional CFA interpolation method that is based on multiscale color gradients[4]. The developed method is applied to Bayer and Lukac patterns with great results which shows that the relationship between gradients at different scales can be a very effective feature to optimally combine directional estimates. This method, the horizontal and vertical color difference estimates are blended based on the ratio of the total absolute values of vertical and horizontal color difference gradients over a local window. The solution does not require any thresholds since it does not make any hard decisions, and it is noniterative. Although designed for the Bayer mosaic pattern[11][12], the proposed method can be modified to be applied to Lukac mosaic patterns. Lukac mosaic pattern is similar to Bayer pattern in the sense that it consists of pure RGB components. When shift every other row in a Bayer pattern by one pixel to either side, obtain the Lukac pattern. Hence, the horizontal relationship between the pixels is still the same, but the vertical arrangement is significantly altered. The performance of the proposed method is evaluated on the test set images. The full color images are first downsampled using Bayer and Lukac patterns. Then, they were reconstructed using the proposed method for each pattern. The difference between the Bayer and the lukac images is reported in terms of PSNR and MSE error measures. The PSNR and MSE comparison results for the Bayer and Lukac pattern are summarized in Table I. Average PSNR is value of bayers pattern is 40.8 and Lukac pattern is 39.38. Two output image produced by using both Bayers and Lukac pattern. Quality of output image from both Bayer and Lukac pattern is compared by using PSNR (Peak signal to Noise Ratio) and MSE (Mean square error) values and found that Bayers Pattern is more efficient compare to Lukac Pattern. Bayers pattern find most missing color pixels so that output image become high quality.
TABLE I
COMPARISON OF PSNR AND MSE FOR BAYER AND LUKAC PATTERN

<table>
<thead>
<tr>
<th>No</th>
<th>Bayer pattern</th>
<th>Lukac pattern</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>PSNR</td>
<td>MSE</td>
</tr>
<tr>
<td>1</td>
<td>42.16</td>
<td>0.98</td>
</tr>
<tr>
<td>2</td>
<td>40.13</td>
<td>1.274</td>
</tr>
<tr>
<td>3</td>
<td>40.36</td>
<td>1.343</td>
</tr>
</tbody>
</table>

Conclusion
Most digital cameras use a color filter array to capture the colors of the scene. Downsampled versions of the red, green, and blue components are acquired, and an interpolation of the three colors is necessary to reconstruct a full representation of the image. This color interpolation is known as demosaicing. Propose a demosaicing method that uses multiscale color gradients to adaptively combine color difference estimates from different directions. The proposed solution does not require any thresholds since it does not make any hard decisions, and it is noniterative. In the proposed system demosaicing based on Bayer and other RGB pattern (eg. Lukac pattern) and Compare the performance in both patterns. Efficiency of each one to be find out. Here A directional CFA interpolation method that is based on multiscale color gradients. The developed method is applied to Bayer and Lukac patterns with great results which shows that the relationship between gradients at different scales can be a very effective feature to optimally combine directional estimates. This method, the horizontal and vertical color difference estimates are blended based on the ratio of the total absolute values of vertical and horizontal color difference gradients over a local window. Two output image produced by using both Bayers and Lukac pattern. Quality of output image from both Bayer and Lukac pattern is compared by using PSNR (Peak signal to Noise Ratio) and MSE (Mean square error) values and found that Bayers Pattern is more efficient compare to Lukac Pattern. Bayers pattern find most missing color pixels so that output image become high quality.

Acknowledgment
The author would like to thank all the teachers and staff of the department of Computer Science and Engineering, TKM Institute of Technology, Karuvelil for their cooperation and support.

References
[4] Ibrahim Pekkucuksen, Yucel Altunbasak “Edge oriented directional color filter array interpolation” Center for Signal and Image Processing, Georgia Institute of Technology, Atlanta, GA, 30308, USA


