Edge Potency Filter Based Color Filter Array Interruption

GURRALA MAHESHWAR
Dept. of ECE

B. SOWJANYA
Dept. of ECE

KETHAVATH NARENDER
Associate Professor,
Dept. of ECE

PRAKASH J. PATIL
Head of Dept.ECE
Vijay Rural Engineering College, Nizamabad, Jntu-H

Abstract:
The basis of the proposed algorithm is the observation that the constant color difference assumption tends to fail across edges. A commercial digital camera captures only one of these channels at each pixel location and the other two needs to be estimated to generate the complete color information this process is called color filter array (CFA) interpolation. Most commercial digital cameras are provided three color sensors (RGB) for quality color images. In order to reduce the cost, the use of one sensor per channel has been avoided with the use of color filter array (CFA) in front of the sensor and then interpolates the missing color samples to obtain a three channel color image. This interpolation introduces special correlations which are likely to be destroyed when tampering with an image. The red and blue images are sampled at a lower rate, so if standard interpolation techniques are used, the reconstructed red and blue images will be missing some high frequency information and could contain distortions from aliasing. This paper proposes an orientation-free edge strength filter and applies it to the demos icing problem. Edge strength filter output is utilized both to improve the initial green channel interpolation and to apply the constant color difference rule adaptively. This simple edge directed method yields visually pleasing results with high CPSNR.

Keywords: *CFA, CPSNR, Edge potency, Filter based color, RGB*

1. Introduction

The growing popularity of digital photography demands every attempt of improvement in terms of quality and speed of the features provided in digital cameras. The heart of a digital still or video camera is its sensor, a 2-D array of Photo sites that measure the amount of light absorbed during the exposure time. The color information is obtained by means of a color filter array (CFA) overlaid on the sensor, such that each photo site is covered by a color filter sensitive to only a portion of the visible light spectrum [7].Color images require multiple data samples for each pixel as opposed to grayscale images for which a pixel is represented by only one data sample. For the RGB image format, these data samples represent red, green and blue channels. A typical digital camera captures only one of these channels at each pixel location and the other two needs to be estimated to generate the complete color information. This process is called color filter array (CFA) interpolation or demosaicing. Although many different CFA patterns have been proposed, the most prevalent one is the Bayer pattern shown in Fig1. As an important step in image processing pipeline of digital cameras, demosaicing has been an area of interest in both academia and industry. The simplest approach to the demosaicing problem is to treat color channels separately and fill in missing pixels in each channel using a spatially invariant interpolation method such as bilinear or bi-cubic interpolation. While such an approach works fine in homogenous areas, it leads to color artifacts and lower resolution in regions with texture and edge structures. Obtaining better demosaicing performance is possible by exploiting the correlation between the color channels. Spectral correlation can be modeled by either constant color ratio rule [2], [3] or constant color difference rule [4], [5]. The basic assumption is that color ratio/difference is constant over a local distance inside a given object. This assumption is likely to break apart across boundaries; hence many demosaicing algorithms try to utilize it adaptively in one way or another. Since the Bayer CFA pattern has twice as many green channel samples as red and blue ones, green channel suffers less from aliasing and is the natural choice as the starting point of the CFA interpolation process. In [6], proposed improving red and blue channel interpolation by adding high frequency components extracted from green channel to red and blue channels. In another frequency-domain approach used an alternating projections scheme based [7]

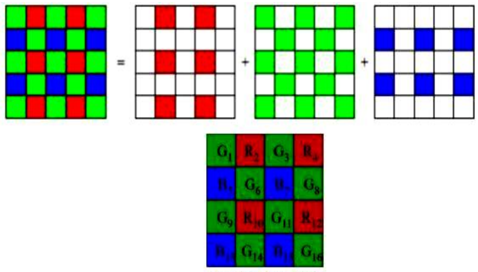
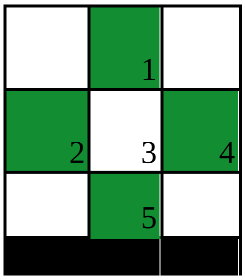


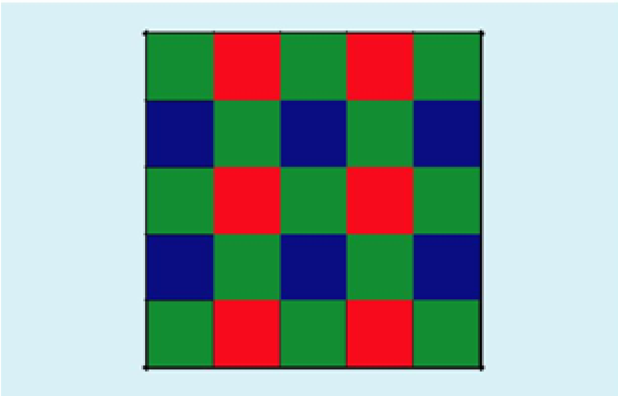
Fig.1 Bayer CFA pattern

on strong inter-channel correlation in high frequency sub bands. Although the main objective is to refine red and blue channels iteratively, the same approach can also improve green channel interpolation (GCI) beforehand which in turn yields better red and blue channel results. A more recent method [8] makes several observations about color channel frequencies and suggests that filtering the CFA image as a whole instead of individual color channels should preserve high frequency information better. To estimate luminance, the method proposes a fixed 5-by-5 filter at green pixel locations and an adaptive filter for red and blue pixel locations. The estimated full resolution luminance is then used to complete missing the chrominance information. Edge-directed green channel interpolation has been proposed early on with various direction decision rules [4], [5], [9], [10]. The method outlined in [4] is particularly noteworthy because it proposed using derivatives of chrominance samples in initial green channel interpolation. Several subsequent demosaicing algorithms made use of this idea. Authors of [11] proposed using variance of color differences as a decision rule [12] proposed making a soft decision to improve the interpolation performance of the original method [4]. In this method [12], color differences along horizontal and vertical directions are treated as noisy observations of the target pixel color difference and they are combined optimally using the linear minimum mean square error estimation (LMMSE) framework. In [13] further improved directional filtering proposed in [12] by introducing scale adaptive filtering based on linear polynomial approximation (LPA). Several methods proposed performing interpolation in both horizontal and vertical directions and making a posteriori decision based on some criteria. In [15] compared local homogeneity of horizontal and vertical interpolation results and in [16] used color gradients over a local window to make the direction decision. In this paper a robust color filter array interpolation technique for color image enhancement is proposed.

2. Image formation process

Since some of the demosaicking methods make explicit use of image formation models, we provide a brief summary of image formation before reviewing the demosaicking methods. The imaging process is usually modeled as a linear process between the light radiance arriving at the camera and the pixel intensities produced by the sensors. Most digital cameras use charge-coupled device (CCD) sensors. In a CCD camera, there is a rectangular grid of electron-collection sites laid over a silicon wafer to record the amount of light energy reaching each of them. When photons strike these sensor sites, electron-hole pairs are generated, and the electrons generated at each site are collected over a certain period of time. The numbers of electrons are eventually converted to pixel values.





Bayer color filter array arrangement

Calculate horizontal gradient $\Delta H = |G2 - G4|$
Calculate vertical gradient $\Delta V = |G1 - G5|$
If $\Delta H > \Delta V$,
$G3 = (G1 + G5)/2$
Else if $\Delta H < \Delta V$,
$G3 = (G2 + G4)/2$
Else
$G3 = (G1 + G5 + G2 + G4)/4$

FIG3] Edge-directed interpolation for the G channel is illustrated. G1, G2, G4, and G5 are measured G values; G3 is the estimated G value at pixel 3.

2.1 Demosaicking Methods

We examine demosaicking methods in three groups. The first group consists of heuristic approaches. The second group formulates demosaicking as a restoration problem. The third group is a generalization that uses the spectral filtering.

2.2 Heuristic Approaches

Heuristic approaches do not try to solve a mathematically defined optimization problem. They are mostly filtering operations that are based on reasonable assumptions about color images. Heuristic approaches may be spatially adaptive, and they may exploit correlation among the color channels. We now overview these heuristic approaches.

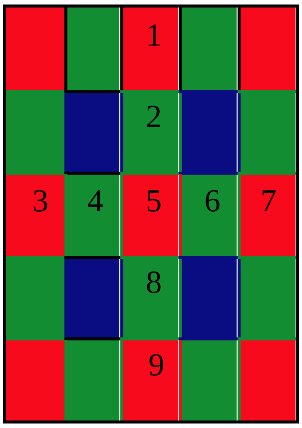
2.3 Edge-Directed Interpolation

Although no adaptive algorithms (e.g., bilinear interpolation or bicubic interpolation) can provide satisfactory results in smooth regions of an image, they usually fail in textured regions and edges. Edge-directed interpolation is an adaptive approach, where the area around each pixel is analyzed to determine if a preferred interpolation direction exists. In practice, the interpolation direction is chosen to avoid interpolating across edges, instead interpolating along any edges in the image.

An illustration of edge-directed interpolation is where horizontal and vertical gradients at the location where G is not measured are calculated from the adjacent G pixels. In [17], these gradients are compared to a constant thresh-old. If the gradient in one direction falls below the threshold, interpolation is performed only along this direction. If both gradients are below or above the threshold, the pixels along both directions are used to estimate the missing value.

The edge-directed interpolation idea can be modified by using larger regions (around the pixel in question) with more complex predictors and by exploiting the texture similarity in different color channels. In [23], the R and B channels (in the 5 × 5 neighborhood of the missing pixel) are used instead of the G channel to determine the gradients. To determine the horizontal and vertical gradients at a B (R) sample, second-order derivatives of B (R) values are used. This algorithm is illustrated in Figure 4. Another example of the edge-directed interpolation is found in [19], where the Jacobian of the R, G, and B samples is used to determine edge directions.

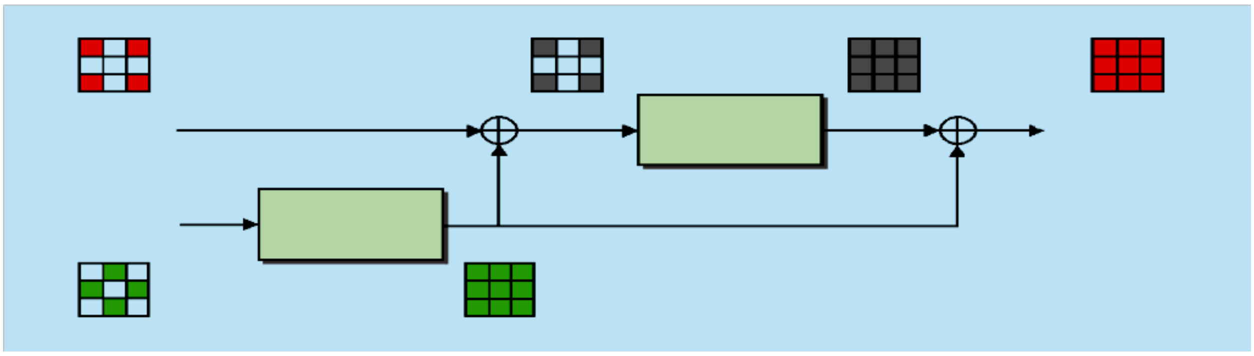
1. Calculate horizontal gradient $\Delta H = \square\ (R3 + R7)/2 - R5$ □
2. Calculate vertical gradient $\Delta V = \square\ (R1 + R9)/2 - R5$ □
3. If $\Delta H > \Delta V$,
   $G5 = (G2 \square G8)/2$
   Else if $\Delta H < \Delta V$,
   $G5 = (G4 \square G6)/2$
   Else
   $G5 = (G2 \square G8 + G4 \square G6)/4$



Edge-directed interpolation in [23] is illustrated for estimating the *G* value at pixel 5. The R values are used to determine the edge direction. When the missing *G* value is at a B pixel, the B values are used to determine the edge direction.

2.4 Weighted Average

In edge-directed interpolation, the edge direction is estimated first, and then the missing sample is estimated by interpolating along the edge. Instead, the likelihood of an edge in a certain direction can be found, and the interpolation can be done based on the edge likelihoods. Such an algorithm was pro-posed by Kimmel in [22]. The algorithm defines edge indicators in several directions as measures of edge likelihood in those directions and determines a missing pixel intensity as a weighted sum of its neighbors. If the likelihood of an edge crossing in a particular direction is high, the edge indicator returns a small value, which results in less contribution from the neighboring pixel in that direction. The G channel is interpolated first; the R and B channels are interpolated from the R-to-G and B-to-G ratios. The color channels are then updated iteratively to obey the constant color ratio rule.



[FIG5] Constant-difference-based interpolation is illustrated for the R channel. The B channel is interpolated similarly

A similar algorithm was proposed recently, where edge indicators are determined in a 7x7 window for the G and a 5x5 window for the R and B channels. In this case, the edge indicator function is based on the $L1$ norm (absolute difference) as opposed to the $L2$ norm]. A related algorithm is proposed, where the directions (horizontal, vertical, diagonal) that have the smallest two interpolation.

3. Proposed algorithm

The basis of the proposed algorithm is the observation that the constant color difference assumption tends to fail across edges. If one can effectively utilize edge information to avoid averaging non-correlated color differences, demosaicing performance could increase dramatically.

The question at this point is, how the edge information can be expressed meaningfully at the pixel level so that it is useful enough to improve demosaicing performance. Edge detection filters such as Sobel and Canny can tell whether an edge structure is present at a given pixel. However, they do not provide any information about the sharpness of luminance transition at that particular pixel. The proposed filter is very useful for finding edges in a grayscale image. However, a mosaicked image only has one of the three color channels available for every pixel location and it certainly does not have complete luminance information at any pixel. That is why, the edge strength filter can only be applied to a mosaicked image by making an approximation. Instead of trying to estimate luminance information and taking estimated luminance differences of neighboring pixel pairs, we take the difference in terms of the available color channel for each pixel pair.

3.1 Green Channel Interpolation

We propose making a hard decision based on the edge strength filter described above. For this purpose, every green pixel to be interpolated (red or blue pixel in the mosaicked image) is marked either horizontal or vertical by comparing the edge strength differences along each direction on a local window

3.2 Green Channel Update

The second step of the proposed algorithm is updating the green channel. We make use of the constant color difference assumption combined with edge strength filter to improve the initial green channel interpolation while avoiding averaging across edge structures. For every green pixel to be updated, the closest four neighbors with available color difference estimates are considered. We expect the edge strength difference between two pixels to be large across edges. That is why the weight for each neighbor is inversely correlated with the total absolute edge strength difference in its direction. In other words, a neighbor will contribute less to the update result if there happens to be a strong edge between the target pixel and itself.

3.3 Red and Blue Channel Interpolation

Once the green channel interpolation is finalized, we fill in red and blue channels using constant color difference assumption. For red channel interpolation at blue pixels and blue channel interpolation at red pixels, diagonal neighbors are used adaptively based on green channel gradients in both directions.

Step1: Themosaicked image has one of the three color channels available for every pixel location and it certainly does not have complete luminance information at any pixel. The edge strength filter (ESF)can be applied to a mosaicked image by making an approximation. Instead of trying to estimate luminance information and taking estimated luminance differences of neighboring pixel pairs, we take the difference in terms of the available color channel for each pixel pair. For instance, for the red center pixel case the diagonal differences will come from the blue channel and the rest from the green channel

$$Spr_{10} = \frac{|pb_5 - pb_{15}|}{2} + \frac{|pb_7 - pb_{13}|}{2} + |pg_6 - pg_{11}| + |pg_9 - pg_{11}| \qquad (1)$$

The edge strength for green and blue pixels will be calculated in the same way. The edge strength map obtained from the mosaicked input image will help us both in initial green channel interpolation stage and in subsequent green channel update.

Step2:In this process every green pixel interpolated is marked either horizontal or vertical by comparing the edge strength differences along each direction on a local window. For a window size of 5 by 5, horizontal and vertical difference costs can be formulated as follows:

$$H_{i,j} = \sum_{m=-2}^{2} \left(\sum_{n=-2}^{1} S_{i+m,j+n} - S_{i+m,j+n+1} \right) \qquad (2)$$

$$V_{i,j} = \sum_{m=-2}^{2} \left(\sum_{n=-2}^{1} S_{i+m,j+n} - S_{i+m+1,j+n} \right)$$

Where $S_{i,j}$ is the edge strength filter output at pixel location (i,j), and $H_{i,j}$andVi,j represent the total horizontal and vertical costs, respectively. The target pixel will be labeled horizontal if horizontal cost is less than vertical and vice versa. The rationale behind this decision scheme is that if there happens to be a horizontal edge in a given

$$\widetilde{pg}_{i,j} = \begin{cases} pb_{i,j} + \frac{\widetilde{pg}^H_{i,j} - pb_{i,j}}{2} + \frac{pg_{i,j-1} - \widetilde{pb}^H_{i,j-1}}{4} + \frac{pg - \widetilde{pb}^H_{i,j+1}}{4}, & \text{if horizontal} \\ pb_{i,j} + \frac{\widetilde{pg}^V_{i,j} - pb_{i,j}}{2} + \frac{pg_{i-1,j} - \widetilde{pb}^V_{i-1,j}}{4} + \frac{pg_{i-1,j} - pb}{4}, & \text{if vertical} \end{cases} \qquad (3)$$

where directional estimations are calculated by

$$\widetilde{pg}^H_{i,j} = \frac{pg_{i,j-1} + pg_{i,j+1}}{2} + \frac{2 * pb_{i,j} - pb_{i,j-2} - pb_{i,j+2}}{4} \qquad (4)$$

$$\widetilde{pg}^V_{i,j} = \frac{pg_{i-1,j} + pg_{i+1,j}}{2} + \frac{2 * pb_{i,j} - pb_{i-2,j} - pb}{4}$$

$$\widetilde{pb}^H_{i,j} = \frac{pb + pb_{i,j+1}}{2} + \frac{2 * pg_{i,j} - pg_{i,j-2} - pg}{4}$$

$$\widetilde{pb}^V_{i,j} = \frac{pb_{i-1,j} + pb_{i+1,j}}{2} + \frac{2 * pg_{i,j} - pg_{i-2,j} - pg_{i+2,j}}{4}$$

Neighborhood, then the edge strength differences between vertical neighbors will vary more than those of horizontal neighbors. After all the pixels are labeled, the robustness of the direction decision can be improved by relabeling them based on the directions of their neighbors. For instance, considering the closest 8 neighbors of a target pixel and the pixel itself, it will be labeled horizontal only if more than 4 of those 9 pixels are initially labeled horizontal. Based on the final direction label, green channel is interpolated as follows:

Step3: In This step we update the three color channels by making use of the constant color difference assumption combined with edge strength filter to improve the initial green channel interpolation while avoiding averaging across edge structures. For every green pixel to be updated, the closest four neighbors with available color difference estimates are considered. We expect the edge strength difference between two pixels to be large across edges. That is why the weight for each neighbor is inversely correlated with the total absolute edge strength difference in its direction. In other words, a neighbor will contribute less to the update result if there happens to be a strong edge between the target pixel and itself. Assuming we are updating the green channel value at a blue pixel:

$$d_1 = |s_{i,j} - s_{i-1,j}| + |s_{i-1,j} - s_{i-2,j}| + |s_{i-2,j} - s_{i-3,j}| + c_1$$

$$d_1 = |s_{i,j} - s_{i-1,j}| + |s_{i-1,j} - s_{i-2,j}| + |s_{i-2,j} - s_{i-3,j}| + c_1 \tag{5}$$

$$d_2 = |s_{i,j} - s_{i,j-1}| + |s_{i,j-1} - s_{i,j-2}| + |s_{i,j-2} - s_{i,j-3}| + c_1$$

$$d_3 = |s_{i,j} - s| + |s_{i,j+1} - s_{i,j+2}| + |s_{i,j+2} - s_{i,j+3}| + c_1$$

$$d_4 = |s_{i,j} - s_{i+1,j}| + |s_{i+1,j} - s_{i+2,j}| + |s_{i+2,j} - s_{i+3,j}| + c_1$$

$$m_1 = d_2 * d_3 * d_4 m_2 = d_1 * d_3 * d_4 m_3 = d_1 * d_2 * d_4 m_4 = d_1 * d_2 * d_3$$

$$m_{Total} = m_1 + m_2 + m_3 + m_4$$

$$pg_{i,j} = pb_{i,j} + w1 * (\widetilde{pg}_{i,j} - pb_{i,j}) + (w2) * [\frac{m_1}{m_{Total}}(\widetilde{pg}_{i-2,j} - pb_{i-2,j}) + \frac{m_2}{m_{Total}}(\widetilde{pg}_{i,j-2} - pb_{i,j-2}) + \frac{m_3}{m_{Total}}(\widetilde{pg}_{i,j+2} - pb_{i,j+2}) + \frac{m_4}{m_{Total}}(\widetilde{pg}_{i+2,j} - pb_{i+2,j})]$$

4. Experimental Results

In this paper, we evaluate the proposed method by several experiments on different database images, and compared the output images of Gaussian with ground truth images by jacquard method, mean and standard deviation and placed in the tables, drawn the graphs and shown the ground truth , noisy and output images. Jacquard similarity coefficient method is very popular and used mostly as similarity indices for binary data. The area of overlap A_j is calculated between the thresholded binary images Bjand its corresponding gold standard image $G_j$as shown below.

$$A_i = \frac{|B_i \cap G_i|}{|B_i \cup G_i|}$$

If the thresholded object and corresponding gold standard image G_j (associated ground truth image) are exactly identical then the measure is 1 and the measure 0 represents they are totally disjoint, but the higher measure indicates more similarity. Jacquard index of the proposed method is compared with the rank filters and the values are shown in Tables and the Figures. Demonstrate the superiority of the proposed method .The proposed method confirms the qualitative improvement over the traditional methods. From the tables and graphs it is clear that in any category of images the mean lies in the range of dominant pixels of the image, or between the two or more dominant pixels region. Hence mean evidences the location of object or back ground. The vagueness in the image is always lies around the edges the edge strengths are represented with standard deviation. The table of values and graph show that the standard deviation always has high values than mean so that the edges are very stronger than the image regions.

4.1 Gaussian noise



Fig.1a Fig.1b Fig.1c

mean	stdv	jaccard
0.3913	4.207	0.7289
0.4507	4.845	0.7372
0.3557	4.824	0.7312
0.5191	5.581	0.6847
0.5346	5.748	0.6808
0.5473	5.884	0.7328
0.5752	6.184	0.8132
0.6698	7.202	0.9177
0.7028	7.556	0.9735
0.6129	6.590	0.9945

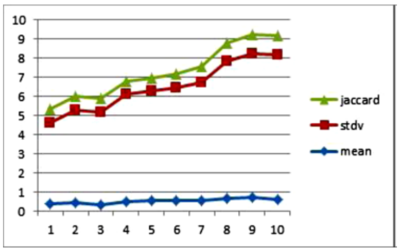


Fig.1d Fig.1e

mean	stdv	jaccard
0.4623	0.5824	0.6041
0.4550	0.5733	0.6694
0.4435	0.5507	0.7311
0.4303	0.5421	0.7930
0.4712	0.5937	0.8556
0.4751	0.5985	0.9101
0.4541	0.5721	0.9557
0.6190	0.7798	0.9839
0.6637	0.8362	0.9958
0.7849	0.9889	0.9986

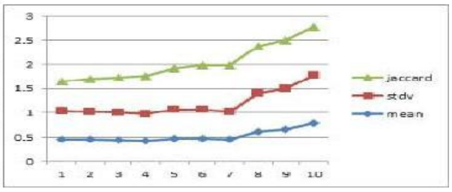


Fig. 2a Fig.2b



Fig.3a Fig.3b Fig. 3c

mean	Stdv	jaccard
0.3515	0.7591	0.5093
0.3971	0.8570	0.5262
0.4297	0.9282	0.5797
0.5115	0.1104	0.6439
0.4688	0.10125	0.7680
0.5490	0.11858	0.8785
0.6191	0.13373	0.9543
0.6321	0.15653	0.9800
0.7488	0.16173	0.9980
0.8122	0.17543	0.9995

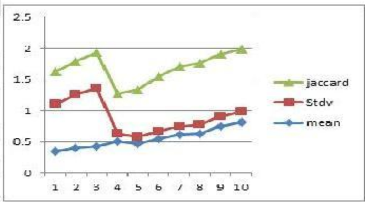


Fig. 3d Fig.3e



Fig.4a Fig.4b Fig. 4c

5. Conclusion

We presented a simple edge strength filter and applied it to the CFA interpolation problem. The edge strength filter helped us identify the regions where constant color difference assumption is likely to fail which in turn lead to improved demosaicing performance. Further research efforts will focus on improving the interpolation results by exploiting spectral correlation more effectively and applying the proposed edge strength filter to other image processing problems.

References

1. Adams, J. E. and Hamilton, J. F. Jr. (1996). "Adaptive color plan interpolation in single sensor color electronic camera," U.S. Patent 5 506 619, Apr. 9.
2. Chung, K. H. and Chan,Y. H.(2006)."Color demosaicing using variance of color differences," IEEE Trans. Image Process., vol. 15, no. 10, pp. 2944–2955, Oct.
3. Hibbard, R. H. (1995). "Apparatus and method for adaptively interpolating a full color image utilizing luminance gradients," U.S. Patent 5 382 976, Jan. 17.
4. Hirakawa, K. and Parks, T. W.(2005). "Adaptive homogeneity-directed demosaicing algorithm," IEEE Trans. Image Process., vol. 14, no. 3, pp. 360–369.
5. Li, X., Gunturk, B. and Zhang, L. (2008). "Image demosaicing: A systematic survey," Proc. SPIE–Int. Soc. Opt. Eng., vol. 6822, p. 68221J-1-15, Jan.
6. Paliy, D. V., Katkovnik, R., Bilcu, S. Alenius and K. Egiazarian (2007)."Spatially adaptive color filter array interpolation for noiseless and noisy data," Int. J. Imag. Syst. Technol., vol. 17, no. 3, pp. 105–122.
7. Tering for color filter array demosaicking," IEEE Trans. Image Process., vol. 16, no. 10, pp. 2515–2525, Oct. 2007.
8. Zhang, L. and Wu, X.(2005). "Color demosaicking via directional linear minimum mean square-error estimation," IEEE Trans. Image Process., vol. 14, no. 12, pp. 2167–2178, Dec.