CCD IMAGE DEMOSAICING USING LOCALIZED CORRELATIONS

RONEN SHER and MOSHE PORAT
Department of Electrical Engineering
Technion – Israel Institute of Technology
Haifa 32000
ISRAEL
sherr@tx.technion.ac.il http://visl.technion.ac.il/~sherr
mp@ee.technion.ac.il http://visl.technion.ac.il/mp

ABSTRACT

A new approach to image interpolation using spatial relationships between adjacent pixels is introduced. In its first stage, the localized statistical relationships are studied based on the sparse version of the image. In the second stage, the governing rules of the image are used to build an interpolated version. The proposed interpolation method is suitable for color single-CCD images for demosaicing purposes. The correlation rule is studied first for each color component separately, then difference images (modified hues) are built to eliminate the color correlation, leading to a smoother reconstructed signal.

Since in Bayer pattern not all the color components are equally represented, the algorithm deals with the major green component differently from the red and blue, using the green as a basis for the whole image reconstruction. Further statistical tools are added to the algorithm to improve the visual results. We compare our method to presently available demosaicing techniques for single-CCD color imaging with the major emphasis on reducing ghost colors and unreal edges. Our conclusion is that the proposed method can significantly improve interpolation and demosaicing tasks in image processing.

Key-Words: CCD Image, Bayer pattern**,** Demosaicing, Least square method.

1. INTRODUCTION

The Bayer pattern has become a common structure in most digital cameras today (Figure 1). "Demosaicing", in which the missing RGB pixels are interpolated, has thus become an integral part of those camera systems. The most basic method for demosaicing is the bi-linear interpolation (BI), which uses the average of the four neighboring pixels in the green plane and two neighbors in the red and blue planes to reconstruct the missing pixel in each color component separately. More complicated reconstruction methods, which work on each color component, are the bi-cubic and the bi-spline that instead of simple average uses polynomials of order 3.

Another group of reconstruction algorithms are based on all the three colors in the neighborhood, however, restrict the computational cost for basic hardware implementation. One of these methods is the constant hue, which is applied as the ratio between red and green or blue to green, the green pixels are reconstructed from BI and then the hue images are reconstructed also with BI. Good results were obtained in [3], which assumes that the ratio between colors in the same pixel is constant, and uses gradients as a weighted coefficient in the reconstruction sum.

The third group of methods has neither component nor computational limitations and is implemented mostly by software. This group uses a non-linear interpolation scheme and includes methods introduced by Kimmel, Gunturk, Muresan, Chung and others. In Kimmel's work [2], the pixels are interpolated along edges. A gradient-based function is built to indicate the edge direction. The green and the ratios of blue/green and red/green (the hues) are interpolated along edges as a weighted sum gradients dependent coefficients. To improve the results, a correction step is performed on the ratios R/G B/G and on the green component using the red and blue values. In Gunturk's work [5], the authors define two types of constraint sets; one imposes consistency with the observed data, and the other arises from the similarity between the high-frequency components of the color channels. An initial estimate is projected onto these constraint sets iteratively until convergence is achieved.

In [6], an improvement compared to [2] is performed by interpolating the difference components rather than the ratios and the interpolation of the green includes a wider neighborhood, e.g., a 15x15 grid. This method results in missing reconstructed data in the periphery.

In [7] the missing green samples are first estimated based on the variances of the color differences along different edge directions. Then the missing red and blue components are estimated based on the interpolated green plane. The neighborhood region is set to 5x5 for edges blocks and to 9x9 for a nonedge blocks thus it also leaves missing reconstructed pixels in the periphery.

Unlike the above methods, in this work we propose a new approach, based on statistical information that represents the image. As could be shown in the next sections, such an approach outperform the existing techniques in many cases.

2. THE ALGORITHM

The new algorithm is based on 4 steps as follows.

Step I

The input data is a CCD image as shown in Figure 1. We first build from the red and the blue components smaller images each, and from the green component two equivalent images (Figure 2).

We start with the green component In_{GR}.. For each pixel $\tilde{p}$ (Figure 3), there are four green neighbors and another four of the red component, denoted $\{n_i\}_{i=1}^{8}$. We want to consider only non-smooth regions in the model, based only on the 4 green neighbors, i.e., $Var\{n_i^+\} > Th_1$, where Th_1 is a variance threshold.

The non-smooth regions are characterized by different edges i.e., vertical, horizontal and diagonal [1]. An edge type is related to the four green neighbors' $\{n_i^+\}_{i=1}^{4}$ intensity values. The model accounts for all the possible permutations of $\{n_i^+\}_{i=1}^{4}$, therefore there are 4!=24 cases/regions i.e.,

(Case 1) $n_1^+ > n_2^+ > n_3^+ > n_4^+$, (Case 2) $n_1^+ > n_2^+ > n_4^+ > n_3^+$, ... (Case 24) $n_4^+ > n_3^+ > n_2^+ > n_1^+$.

For each pixel that belongs to a non-smooth region, the four surrounding green neighbors and the four red neighbors, as well as the concerned pixel are stored in an array A_{GR}., The entry to A_{GR} is the region/case index r, $(r = 1,2,...,24)$ according to the four surrounding green neighbors $\{n_i^+\}_{i=1}^{4}$ permutations in the region.

The same holds for In_{GB}, except for the blue neighbors, which here replace the red.

From those arrays we find the approximated linear relation in each case. Our assumption is that each pixel can be reconstructed as the weighted sum of its neighbors

$$\tilde{p} = \sum_i \alpha_i n_i , \qquad (1)$$

and in each case a different set of coefficients will be used i.e., $\underline{\boldsymbol{\alpha}}_r = [\alpha_1, \alpha_2, ..., \alpha_8]_r$. To find the 24 sets of coefficients $\{\underline{\boldsymbol{\alpha}}_r\}_{r=1}^{24}$, we use the relations stored in A_{GR} based on the original image.

For each case we write Eq. (1) several times, according to the number of times that the case occurred, denoted Q:

$$\underbrace{\begin{bmatrix} p_1 \\ p_2 \\ \vdots \\ p_Q \end{bmatrix}}_{\underline{\mathbf{P}}_r^x} = \underbrace{\begin{bmatrix} n_{11} & n_{12} & \cdots & n_{18} \\ n_{21} & n_{22} & \cdots & n_{28} \\ \vdots & \ddots & & \vdots \\ n_{Q1} & n_{Q2} & \cdots & n_{Q8} \end{bmatrix}}_{\underline{\underline{\mathbf{n}}}_r} \cdot \underbrace{\begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_8 \end{bmatrix}}_{\underline{\boldsymbol{\alpha}}_r} \qquad (2)$$

Or in vector form

$$\underline{\mathbf{P}}_r^x = \underline{\underline{\mathbf{n}}}_r \cdot \underline{\boldsymbol{\alpha}}_r . \qquad (3)$$

Since there are usually more rows than columns in this matrix, we use the least square method for each case:

$$\underline{\boldsymbol{\alpha}}_r = \left[\underline{\underline{\mathbf{n}}}_r^T \cdot \underline{\underline{\mathbf{n}}}_r \right]^{-1} \cdot \underline{\underline{\mathbf{n}}}_r^T \cdot \underline{\mathbf{P}}_r^x . \qquad (4)$$

At this point we have the relation for the green components In_{GR} and In_{GB}. This rule is used to reconstruct the missing pixels in the sparse green component of the CCD image according to (1), while in smooth regions, where $Var\{n_i\} \le Th_1$, a plain average of the four green neighbors serves as reconstruction.

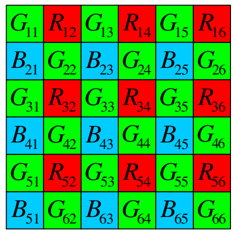

Figure 1 – Bayer pattern.

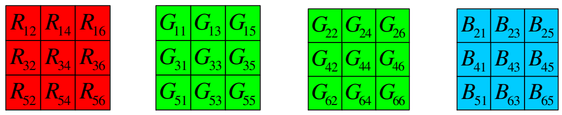

Figure 2 – From left to right: In_R , In_{GR} , In_{GB} , In_B.

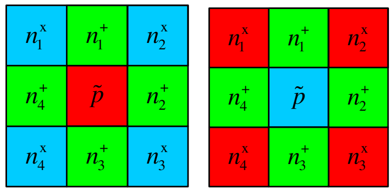

Figure 3 – Green reconstruction region

Step II

We use the reconstructed green component denoted $\tilde{G}$ to reconstruct the red and the blue components. We first subtract the red and blue from the green, i.e., define $\Delta_{RG}^1 = R_{CCD} - \tilde{G}$, $\Delta_{BG}^1 = B_{CCD} - \tilde{G}$. This is done to reduce the variance and further process smoother signals, which can be viewed as modified hues [5].

The difference signals Δ_{RG}^1, Δ_{BG}^1, are naturally not fully populated - the missing red and blue pixels account for 75% of the pixels considered. In this step we wish to reconstruct 25% of the data, those are noted as x-type pixels (Figure 3). If i and j are the row and column indexes, respectively, then the x-pixels locations in the blue component satisfy

$$(\mathrm{mod}(i,2) \neq 0) \, AND \, (\mathrm{mod}(j,2) = 0) , \qquad (5)$$

or simply the locations of the known red pixels. In the red component, the x-pixels satisfy

$$(\mathrm{mod}(i,2) = 0) \, AND \, (\mathrm{mod}(j,2) \neq 0) , \qquad (6)$$

or the locations of the known blue pixels. As in the first step, we build two smaller versions with half the size denoted $\delta_{RG}^1, \delta_{BG}^1$, which are the samples at the red and blue known pixels' positions in Δ_{RG}^1 and Δ_{BG}^1, respectively, excluding missing pixels. The smoothness effect as a result from the subtraction can be observed in Figure 4.

We now use the same technique as for the green reconstruction except that only four neighbors participate in the reconstruction, i.e., (2) becomes

$$\begin{bmatrix} p_1 \\ p_2 \\ \vdots \\ p_Q \end{bmatrix} = \begin{bmatrix} n_{11} & n_{12} & \cdots & n_{14} \\ n_{21} & n_{22} & \cdots & n_{24} \\ \vdots & \ddots & & \vdots \\ n_{Q1} & n_{Q2} & \cdots & n_{Q4} \end{bmatrix} \cdot \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_4 \end{bmatrix}. \qquad (7)$$

The results are two difference images $\Delta_{RG}^2, \Delta_{BG}^2$ with reconstructed pixels in the 'x' spots.

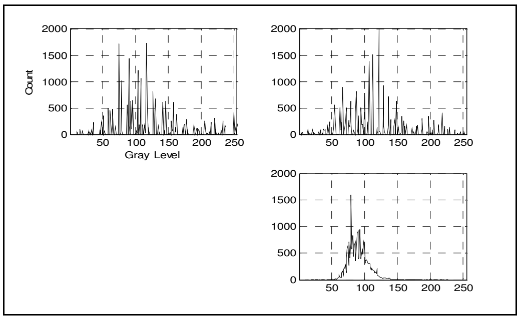

Figure 4 – Components histograms: Top Left - Red (In_R), Top Right - Reconstructed green in the red locations (In_{GR}), Bottom - The difference of the previous two (δ_{RG}^1). Minimization of the variances can be easily observed.

After the primary reconstruction, we further improve the results. The images $\Delta_{RG}^2, \Delta_{BG}^2$ have the same number of pixels, in the same locations. We now subtract them, $\Delta_{RB} = \Delta_{RG}^2 - \Delta_{BG}^2$, to provide in each location a difference between red and blue. In half of these cases, the red is reconstructed; in the other half we calculate the blue.

At this point we start a second pass to reconstruct the pixels in the 'x' location. Note that in the first pass we have studied the relations between the pixels where the blue is reconstructed and the red is the original CCD value, i.e., satisfying (5). The pixel in the left hand side vector of (7) is populated by the pixels in locations where the red is reconstructed and the blue is the original. We use the concluded relations from the second half where the blue is the original and the red is reconstructed, satisfying (6) to construct the image Δ_{RB}^1. This iteration reduces the noise and projects the relations from the red onto the blue in the difference image. The next iteration is the complementary situation – i.e., the blue is the original, used with half of the original red. This step provides a difference reconstructed image Δ_{RB}^2.

To return to colors coordination, we use the following:

$$\begin{cases} if\,(5) \text{ then } \tilde{R}_1 = R_{CCD} \\ if\,(6) \text{ then } \tilde{R}_1 = \tilde{\Delta}_{RB} + B_{CCD} \end{cases} \qquad (7)$$

$$\begin{cases} if\,(5) \text{ then } \tilde{B}_1 = R_{CCD} - \tilde{\Delta}_{RB} \\ if\,(6) \text{ then } \tilde{B}_1 = B_{CCD} \end{cases} \qquad (8)$$

Step III

We reconstruct the additional 50% missing pixels in the red and the blue planes. Instead of working on the colors planes, we use here again the difference images, $\Delta_{RG}^3 = \tilde{R}_1 - \tilde{G}$, $\Delta_{BG}^3 = \tilde{B}_1 - \tilde{G}$. The first pass in this step is building the smaller images δ_{RG}^3, δ_{BG}^3 in the known pixels from Δ_{RG}^3, Δ_{BG}^3 i.e., satisfying (5) or (6). From δ_{RG}^3, δ_{BG}^3 we conclude the correlation rule and use it to reconstruct the missing pixels in the locations:

$$\left(\mathrm{mod}(i,2) = 0\right) AND \left(\mathrm{mod}(j,2) = 0\right) \qquad (9)$$

or

$$\left(\mathrm{mod}(i,2) \neq 0\right) AND \left(\mathrm{mod}(j,2) \neq 0\right) \quad . \qquad (10)$$

The outputs are Δ_{RG}^4, Δ_{BG}^4. The second pass is needed to improve the results and emphasize edges. From the reconstructed pixels in Δ_{RG}^4, Δ_{BG}^4 (at locations according (9) or (10)) we study the rule for the input pixels (present in Δ_{RG}^3, Δ_{BG}^3, satisfying (5) or (6)). Here the left-hand side vector in (7) is populated by the input pixels and the neighbors in the matrix are reconstructed in the first pass. This rule is used for the input pixels to reconstruct again the pixels satisfying (9) or (10). This pass preserves the original rule that holds in the image and compensates the smoothness effect of the first pass reconstruction.

Step IV

To return to the RGB components, we use the difference images and the reconstructed green. The red and blue images are formed by $\tilde{R}_2 = \Delta_{RG}^4 + \tilde{G}_1$, $\tilde{B}_2 = \Delta_{BG}^4 + \tilde{G}_1$. However, the green component will be modified by the data obtained from the red and blue reconstructions, since the first green reconstruction shows artificial shapes near edges. Thus near edges (high gray level variance) the reconstruction is based on Δ_{RG}^4, Δ_{BG}^4 by:

$$\begin{cases} \text{if } \left((9) \text{ OR } (10)\right) \text{ then } \tilde{G}_2 = G_{CCD} \\ \text{else } \begin{cases} \text{if } \left(Var\left\{ \tilde{G}_{i+k,j+l} \right\}_{k=-1,l=-1}^{k=1,l=1} < Th_2 \right) \text{ then } \tilde{G}_2 = \tilde{G}_1 \\ \text{else } \begin{cases} \text{if}\,(5) \text{ then } \tilde{G}_2 = \Delta_{RG}^4 - \tilde{R}_2 \\ \text{if}\,(6) \text{ then } \tilde{G}_2 = \Delta_{BG}^4 - \tilde{B}_2 \end{cases} \end{cases} \end{cases} \qquad (11)$$

where Th_2 is the threshold for the variance. The output image is then composed from $\tilde{R}_2$, $\tilde{G}_2$, $\tilde{B}_2$.

3. STATISTICAL GENERALIZATION

The goal is to divide regions to sub-regions by statistical criteria in order to improve the reconstruction result. Our preliminary knowledge on the region is from the available neighbors, thus their statistical characteristics are helpful for sub-regions division. We use the mean μ and the standard deviation (STD) σ of the surrounding neighbors as the division criteria, according to the following:

$$\mu = \frac{1}{N}\sum_{i=1}^{N} n_i, \qquad (12)$$

$$\sigma = \sqrt{\frac{1}{N-1}\sum_{i=1}^{N}\left(n_i - \mu\right)^2}, \qquad (13)$$

where N is the number of known neighbors (8 or 4).

The mean value μ of the green component will differ between gray level regions, while in the difference images, the μ will differ between regions with different hues. A high value of σ indicates a significant edge, while a low σ indicates a smooth area.

The mean and STD histograms can be easily observed to be invariant to resolution changes (Figs. 5 and 6). The complete algorithm is similar to the process of Section 2, however, with modification to the permutation procedure before obtaining (2) and (7). In each array we examine each region and divide it into sub-regions according to its statistical moments. The reference values for each sub-region edges are studied from the small image ensuring equal size sub-regions. For each quantized sub-region, (2) and (7) are formed to calculate the coefficients, while (1) is used in the reconstruction step.

In the results shown in Figure 7 (image size is 384x256) the division is into three mean and two STDs for the green component (eight neighbors) and seven mean and two STDs for the difference images (four neighbors) sub-regions provide the best visual results. Consequently, we get $24 \cdot 2 \cdot 3 = 144$ cases for the green component and $24 \cdot 2 \cdot 7 = 336$ cases for the difference images.

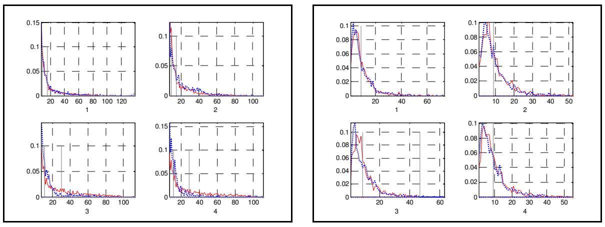

Figure 5 – Image STD histograms for the first four regions (out of 24). Left - Green components; Right - Δ_{RB}. The x-axis is STD values. In red - small image, in blue - the large image; the gray vertical lines are the limits of the equal size STD sub-regions.

4. RESULTS

We have tested the new algorithm on several commonly analyzed images, and compared them to five other known techniques considered to be efficient: Bilinear, Kimmel, Gunturk, Optimal Recovery, and Variance of Color Differences. The numerical values, which were selected experimentally to give the best visual results, are Th_1=7 for the green reconstruction, Th_1=1 for the delta images, and Th_2=12 for the green modification. Zoom-in areas were selected to illustrate the differences between the various methods. The results for the Lighthouse are shown in Figure 7. Similar results have been obtained for additional commonly used images of “Window”, “Statue” and “Sails” using these six demosaicing methods. Table 1 summarizes the S-CIELab test results [8], [9] for errors greater than 20, where the proposed method has achieved the best score.

5. CONCLUSIONS

We have introduced a new approach to CCD image demosaicing using statistical structural relations based on the partial sparse components of the masked image. This non-linear localized method preserves the relation studied from the input image and overcomes the aliasing effect produced by the CCD Bayer sampling method. We also use a de-correlation tool in the form of difference image to allow a straightforward division into neighborhood cases. The results show less ghost colors and edge preserving compared to presently available methods. Our conclusion is that the new approach to color demosaicing could be very efficient for digital cameras based on Bayer masks.

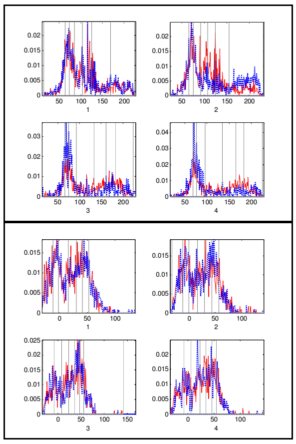

Figure 6 – Mean histograms for the first four regions. Top - Green components; Bottom - Δ_{RB}. The x-axis on the left is the intensity (gray) level. On the right is the ‘hue’ level. In red - small image, in blue - the big image; the gray vertical lines are the limits of the equal size mean sub-regions.

REFERENCES

[1] R. Sher and M. Porat, “Image Interpolation using Consistent Neighborhood Correlations,” Proc. of *ISTASC '05*, Malta, 2005.

[2] R. Kimmel, “Demosaicing: Image Reconstruction from Color CCD Samples,” *IEEE Trans. on Image Processing*, Vol. 8, No. 9, pp. 1221-1228, 1999.

[3] J. E. Adams, “Interactions Between Color Plane Interpolation and other Image Processing Functions in Electronic Photography,” *SPIE*, Vol. 2416 pp.144-151. 1995.

[4] E. Chang, S. Cheung, and D. Y. Pan, “Color Filter Array Recovery using a Threshold-based Variable Number of Gradients,” *Proc. SPIE*, vol. 3650, pp. 36-43, Mar. 1999.

[5] B. K. Gunturk, Y. Altunbasak, and R. M. Mersereau, “Color plane interpolation using alternating projections,” *IEEE Trans. on Image Processing*, 11(9): 997-1013 2002.

[6] D. D. Muresan and T. W. Parks, “Optimal Recovery Demosaicing,” *Proc. IASTED Signal and Image Processing*, pp. 260—265, Aug. 2002.

[7] K.-H. Chung and Y.-H. Chan, “Color Demosaicing using Variance of Color Differences,” IEEE Trans. on Image Processing, vol. 15, no. 10, pp. 2944–2955, 2006.

[8] E. Bennett, M. Uyttendaele, L. Zitnick, R. Szeliski and S. B. Kang, “Video and Image Bayesian Demosaicing with a Two Color Image Prior,” European Conf. on Computer Vision, vol. 1, pp. 508-521, May 2006.

[9] http://white.stanford.edu/~brian/scielab/

Figure 7 – Comparison of the various methods. Top: Original "Lighthouse" with two areas marked for zoom-in comparison. Middle row: Details of the fence reconstructed by (left to right) - Bilinear, Kimmel, Gunturk, Optimal Recovery, Variance of Color Differences and by our new algorithm. Bottom: Details of the house, reconstructed by the above methods (same order from left to right as for the fence).

Table 1 – S-CIELab results for error >20

Method / Region	Bilinear	Kimmel [2]	Gunturk [5]	Optimal Recovery [6]	Variance Differences [7]	Proposed Method
LightHouse	719	517	360	3737	5570	426
Window	130	292	275	5242	5350	25
Statue	131	123	530	5	2357	251
Sails	136	108	413	33	6399	145
average	279.00	260.00	394.50	2254.25	4919.00	**211.75**