LMMSE FREQUENCY MERGING FOR DEMOSAICKING

Weiran Tang, Oscar C. Au, Xing Wen, Yi Yang and Lu Fang

Hong Kong University of Science and Technology
Department of Electronic and Computer Engineering
Clear Water Bay, Kowloon, Hong Kong
{tangweir, eeau,wxxab, yyang, fanglu}@ust.hk

ABSTRACT

For raw images captured by most digital cameras, every pixel has only on color in R, G and B. Kinds demosaicking algorithms are proposed for interpolating the missing two colors. In this article, the relationships inter and intra color channels are analyzed, and basing on the features, we propose a method to divide raw images into sub images and merge them in frequency domain with linear combination. Optimal weights are calculated with estimation values and raw values based on minimum mean square error criteria. Experiments results with different estimations are presented and discussed.

***Index Terms*—** Demosaicking, LMMSE, Frequency Merging

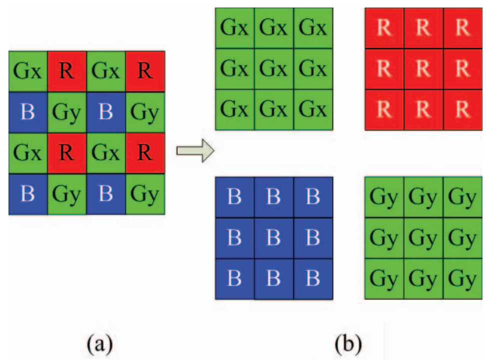

Fig. 1. Bayer CFA Image and Down Sampled Image

1. INTRODUCTION

For reason of cost and size, most digital cameras have only one CCD. So color filter array (CFA) is widely used, such as Bayer filter. For raw images have only one color value at every pixel, demosaicking algorithms are used for recovering missing color values. Detecting edges and interpolating along edges is a very popular idea, and some algorithms [1][3] achieve very good performance. A review and comparison of demosaicking algorithms is given by[4]. Different to processing every pixels, in this article, we will divide the raw image into four sub images, and analyze features of distortions in frequency domain. Basing on the results, one algorithm of the frequency merging with linear minimum mean square error is proposed.

This article is organized as follow: section two is discussion of features; section three is the detail of the LMMSE frequency merging algorithm; experiment results and more discussions are given in section four.

2. FEATURE ANALYSIS IN FREQUENCY DOMAIN

2.1. Dividing and Transforming

For comparison, we assume we have raw images and original images in this section. As mentioned, the raw images captured by digital cameras have only one color at one pixel, as (a) of Figure 1, in which G_x and G_y are both G pixels at different positions. The whole raw image is divided into four raw sub images, as (b) of Figure 1, each with single color and in quarter size of raw image, named with G_{gx}, G_{gy}, R_r and B_b. The capitalizations are used to mark original color values, and the subscripts mean the position in raw image. So G_{gx} means original sub image of G value at G_x pixels. Meantime, lowercases are used for marking interpolation result sub images, such as g_r, which may have distortions to original values.

For the raw image, there are eight missing sub images, two for G channel and three for each R or B channels. Because the features of them are very similar, we just choose sub image of G values at R pixels for later discussion. For comparison, the original image with three whole color channels is also divided into 9 sub images, and original sub images G_r are picked. Then, 2D-DCT transformation is applied to all sub image and transforms them into frequency domain, named with $\hat{G}_{gy}$, $\hat{G}_{gx}$, $\hat{R}_r$, $\hat{B}_b$ and $\hat{G}_r$.

2.2. Comparing and Analysis

The distortion images between frequency raw sub images ($\hat{G}_{gy}$, $\hat{G}_{gx}$, $\hat{R}_r$, $\hat{B}_b$) and frequency original sub images ($\hat{G}_r$) are calculated with formula 1, marked with E_{gx}, E_{gy}, E_r and E_b. Absolute values are calculated, for negative or positive

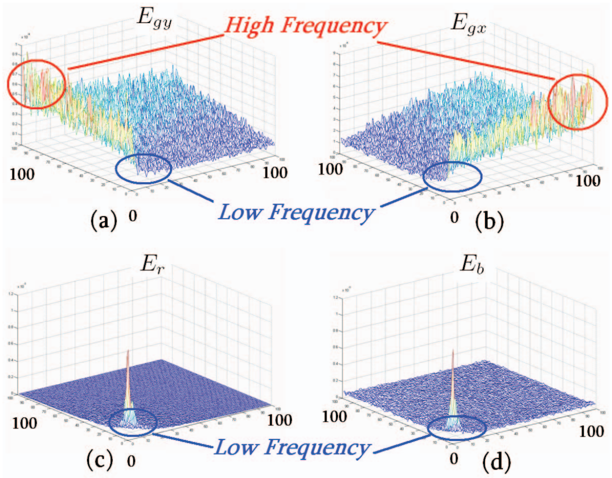

Fig. 2. Difference Between Raw and Original Sub Images

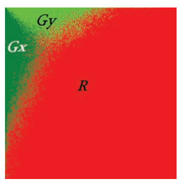

Fig. 3. Furthermore Comparison Image

are meaningless.

$$\begin{aligned} E_{gx} &= |\hat{G}_{gx} - \hat{G}_r|; \\ E_{gy} &= |\hat{G}_{gy} - \hat{G}_r|; \\ E_r &= |\hat{R}_r - \hat{G}_r|; \\ E_b &= |\hat{B}_b - \hat{G}_r|; \end{aligned} \tag{1}$$

All distortion images are still in squarer size of original image. For observing statical features, we scale all distortion images into fixed size, apply this process to all images, and sum up all results. The summary are presented in mesh diagrams, in Figure 2.

In Figure 2, low frequency areas are marked with blue circles, and high frequency with red circles. From (a) and (b) of Figure 2, we can observe that the distortions of neighbor G pixels (E_{gx}, E_{gy}) have smaller values at low frequency area than high frequency area. At the meantime, from E_r and E_b in (c) and (d) of Figure 2, inverse features could be observed: distortions of E_r or E_b at high frequency area are smaller than low frequency area.

Besides Figure 2, more comparison about distortion images is applied. We crate a blank matrix in quarter size of original image. For every pixel in blank matrix, we select the one with minimum value in summary of E_{gy}, E_{gx}, E_r and E_b, and mark this pixel with this color. $\hat{G}_{gy}$ and $\hat{G}_{gx}$ in this matrix are marked with green in different luminance for distinguishing. The result matrix is presented in Figure 3. The left-up corner is the zero point. Similar to previous discussion, G_{gx} and G_{gy} take the low frequency area, and R_r take all high frequency area. Rarely blue point appearing in the matrix presents the feature that $\hat{B}_b$ always has bigger distortion to $\hat{G}_r$ at every frequency area.

At last, we can have two observations: for G_r (G values at R pixels):

1) The G values at neighbor pixels have smallest distortions in low frequency area.

2) The values of red at same pixels have smallest distortions in high frequency area.

Indeed, the first observation means neighbor values in same color channel take low frequency information of G_r. This is the principle of bilinear interpolation algorithm, which just copies low frequency information from neighbor G pixels and causes information losing at high frequency. The second observation can be explained by the high correlations among three color channels of one pixel, which is discussed and used in POCS[2] method. These two features could be observed from other sub images in all three color channels.

3. INTERPOLATION ALGORITHM WITH LMMSE

3.1. Main Idea

In this section, we only have raw image to interpolate. Basing on the analysis in previous section, we propose a new demosaicking algorithm. The main idea is dividing the raw image into four sub images, as (b) of Firure1, and then, merge them together in frequency domain with linear combination. Because four raw sub images have different distortion to missing image in different frequency areas, we hope to control the process of merging exactly: we want to find the optimal weights for each raw sub images in every frequency area. For example, when taking $\hat{G}_r$ for interpolating target, $\hat{R}_r$ should be given more weight in high frequency area, and $\hat{G}_{gx}$ should be given more weight in low frequency area. So the process of linear combination is applied block by block, and weights are calculated for every block. Formula 2 is the expression of the linear combination.

$$\begin{cases} \hat{c}_n = w_1 D_1 + w_2 D_2 + w_3 D_3 + w_4 D_4; \\ 1 = w_1 + w_2 + w_3 + w_4 \end{cases} \tag{2}$$

The result $\hat{c}_n$ are blocks in missing sub images. c means color and n means positions of pixels. D_i (D_1, D_2, D_3, D_4) are blocks in four raw sub images: $\hat{R}_r$, $\hat{G}_{gx}$, $\hat{G}_{gy}$ and $\hat{B}_b$. w_1, and w_2, w_3 and w_4 are the weights for each one.

For finding optimal weights, we hope $E[\hat{O}_n - \hat{c}_n]$, the expectation of distortion between interpolation result $\hat{c}_n$ and original values $\hat{O}_n$, to be minimized. In practical, instead of original values, we use estimation values $\hat{O}'_n$, which could be result of any other demosaicking algorithm, and we minimize $E[\hat{O}'_n - \hat{c}_n]$ instead of $E[\hat{O}_n - \hat{c}_n]$. We will discuss the difference between using original values and using estimation values later.

For using result of other algorithms, this algorithm proposed by us could be considered as an enhancement method. This method could always improve the performance of estimation values in one or all color channels, although choosing different algorithm as estimation values leads different performance of result.

3.2. Optimal Weights

Finding optimal weights in the linear combination is the key of this algorithm. v_i are distortions between estimation sub images $\hat{O}'_n$ and raw sub images D_i, as formula 3,

$$\begin{aligned} v_i &= \hat{O}'_n - D_i \\ D_i &= \hat{O}'_n - v_i \end{aligned} \tag{3}$$

Substitute D_i in formula 2 with D_i in formula 3, the distortion between estimation values and results of linear combination can be rewritten as:

$$\begin{aligned} \xi &= \hat{O}'_n - \sum w_i D_i \\ &= \hat{O}'_n - w_1 D_1 - w_2 D_2 - w_3 D_3 - w_4 D_4 \\ &= w_1(v_1 - v_4) + w_2(v_2 - v_4) + w_3(v_3 - v_4) + v_4 \end{aligned} \tag{4}$$

For finding optimal weights based on minimum mean square error criteria, we minimum $E[\xi^2]$. Therefore we differentiate $E[\xi^2]$ respect to each w_i as follow.

$$\begin{aligned} E[(v_1 - v_4) * \xi] &= 0 \\ E[(v_2 - v_4) * \xi] &= 0 \\ E[(v_3 - v_4) * \xi] &= 0 \end{aligned} \tag{5}$$

Solving the equations in Formula5, we have the final formulas for weights:

$$\begin{pmatrix} w_1 \\ w_2 \\ w_3 \end{pmatrix} = M * \begin{pmatrix} E[v_4^2 - v_1 v_4] \\ E[v_4^2 - v_2 v_4] \\ E[v_4^2 - v_3 v_4] \end{pmatrix} \tag{6}$$

$$M = \begin{pmatrix} e(1,4,1) & e(1,4,2) & e(1,4,3) \\ e(1,4,2) & e(2,4,2) & e(2,4,3) \\ e(1,4,3) & e(2,4,3) & e(3,4,3) \end{pmatrix}^{-1} \tag{7}$$

$$e(i,j,k) = E[(v_i - v_j)(v_k - v_j)] \tag{8}$$

The expectations in formula6 and formula8 are calculated as follow:

$$E[v_j^2 - v_i v_j] = \sigma_j^2 + m_j^2 - \sigma_i \sigma_j \rho_{ij} - m_i m_j \tag{9}$$

For $E[(v_i - v_j)(v_k - v_j)]$, when $i = k$:

$$E[(v_i - v_j)^2] = \sigma_i^2 + \sigma_j^2 - 2\sigma_i \sigma_j \rho_{ij} + (m_1 - m_j)^2 \tag{10}$$

For $E[(v_i - v_j)(v_k - v_j)]$, when $i \neq k$:

$$\begin{aligned} E[(v_i - v_j)(v_k - v_j)] = {} & \sigma_i \sigma_k \rho_{ik} + m_i m_k + \sigma_j^2 + m_j^2 \\ & - \sigma_i \sigma_j \rho_{ij} - m_i m_j - \sigma_i \sigma_k \rho_{ik} - m_i m_k \end{aligned} \tag{11}$$

In forumla9- forumla11, σ_i is the variance of v_i, ρ_{ik} is the correlation coefficient between v_i and v_k, and m_i is the mean of v_i. So we can find that only distortions v_i are used in calculating optimal weights.

3.3. Process Steps

The main steps of this algorithm are as follow:

1) Divide raw image and estimation image into four raw sub images (G_{gx}, G_{gy}, R_r, B_b), and twelve estimation sub images O'_n, each in quarter size of original image, as (b) of Figure 1. All sub images are transformed into frequency domain with 2D-DCT transformation.

2) For interpolate one missing sub image, we pick the correspondence estimation sub image. Choose block size and calculate distortion images v_i with formula3.

3) Calculate parameters of v_i: mean values m_i, variance values σ_i, and correlation coefficients ρ_{ij}. With m_i, σ_i and ρ_{ij}, expectations in formula9, 10 and 11 are calculated. After calculating matrix M in formula 7, vector of weights in formula6 is obtained.

4) Using weights, blocks in the missing sub images are interpolated from blocks of four raw sub images with formula 2. After all blocks are interpolated, result sub image is transformed into spatial domain with inverse 2D-DCT transformation.

5) Repeat step(2) to (4) for every missing sub image. For G channel, we need interpolate two missing sub images: g_r and g_b, and for R or B channels, three sub images are needed for each.

4. EXPERIMENT RESULTS

In experiments, we use results of different demosaicking algorithms as estimation values: bilinear, PSCD[1] and POCS[2]. The images used in experiment are shown in figure 4. Table 1 are PSNR of estimation values and results of FM (frequency merging method which we proposed). From this tables, we can find that proposed method improves result at one or all color channels: compared to POCS or bilinear method, FM improves result for every channel; compared to PCSD, FM improves G channels for every image. And we also find that results of FM with bilinear as estimation have better performance than the other two at G channel, although bilinear algorithm gives worst result.

5. CONCLUSION AND MORE DISCUSSION

In theory, we need original values for LMMSE, but in practical we only have estimation values. Because weights are only

Fig. 4. Images used in Experiment

calculated from parameters (m_i, σ_i, ρ_{ij}) of distortions v_i, if the parameters of distortions between estimations values and raw images are similar to parameters of distortions between original values and raw images, we will have good interpolation result approaching to original values. So the performance of FM algorithm is decided by whether the estimation values make estimation parameters (parameters of distortions between raw sub images and estimation values) similar to real parameters (parameters of distortions between raw sub images and original values). This is also the explanation of why FM with bilinear as estimation have best result in G. Till now, for time limit, we have not test every algorithm.

6. ACKNOWLEDGEMENT

This work has been supported in part by the Innovation and Technology Commission of the Hong Kong Special Administrative Region, China (project no. GHP/048/08).

7. REFERENCES

[1] Wu, X. and Zhang, N. ”Primary-consistent soft-decision color demosaicking for digital cameras” *IImage Processing, IEEE Transactions on*, vol. 13, no. 9, pp. 1263–1274, 2004.

[2] Gunturk, BK and Altunbasak, Y. and Mersereau, RM ”Color plane interpolation using alternating projections” *Image Processing, IEEE Transactions on*, vol. 11, no. 9, pp. 997–1013, 2002.

[3] Yuk, C.K.M. and Au, O.C. and Li, R.Y.M. and Lam, S.Y. ”Primary-consistent soft-decision color demosaicking for digital cameras” *Multimedia Signal Processing, 2007. MMSP 2007. IEEE 9th Workshop on*, pp. 449–452, 2007.

[4] Gunturk, BK and Glotzbach, J. and Altunbasak, Y. and Schafer, RW and Mersereau, RM ”Demosaicking: color filter array interpolation” *Signal Processing Magazine, IEEE*, vol. 22, no. 1, pp. 44–54, 2005.

Table 1. PSNR of Experiment Results

Using Bilinear as FM’s Estimation						
	G		R		B	
Image	FM	bilinear	FM	bilinear	FM	bilinear
1	41.65	29.39	27.77	24.91	28.11	25.11
2	44.00	35.55	31.99	31.54	32.87	30.69
3	42.11	30.77	28.67	26.32	28.68	26.48
4	37.18	27.30	25.62	22.42	26.10	22.62
5	39.99	33.01	31.72	29.60	31.68	28.61
6	41.27	32.04	30.25	27.94	31.02	28.05
7	38.53	26.41	25.82	22.98	25.77	23.01
8	45.16	34.11	31.90	29.73	31.25	29.25
9	37.63	29.09	28.59	26.09	27.59	24.59
10	41.16	31.28	30.15	27.40	29.92	26.98
11	43.01	34.84	33.50	30.83	33.78	30.55
12	42.10	34.20	34.16	30.77	33.30	30.18
13	38.26	30.34	29.55	26.73	29.87	26.96
14	37.83	31.74	28.95	27.94	29.56	27.87
15	41.54	31.47	29.96	26.73	30.21	26.78
16	42.12	33.32	31.95	29.13	32.25	29.51
Using PCSD as FM’s estimation						
	G		R		B	
Image	FM	PCSD	FM	PCSD	FM	PCSD
1	38.86	35.41	34.98	32.88	35.96	33.01
2	42.92	41.28	33.97	36.22	39.50	38.32
3	41.00	37.68	37.09	35.04	36.81	34.64
4	35.81	33.99	31.46	31.18	32.09	31.12
5	39.41	38.04	34.87	35.44	35.19	34.96
6	39.98	37.89	35.15	35.26	37.09	35.84
7	35.40	31.37	32.74	29.44	32.61	29.35
8	44.20	41.58	40.39	38.94	39.96	38.70
9	36.36	34.22	33.32	32.30	32.11	30.83
10	39.76	36.85	36.01	34.63	35.99	34.14
11	41.82	41.74	37.45	38.93	38.48	39.26
12	40.85	39.88	37.77	38.12	37.86	38.06
13	36.96	34.94	33.08	32.90	34.91	33.15
14	37.29	37.35	31.58	34.16	33.22	34.58
15	40.53	38.64	35.94	35.68	37.78	36.23
16	41.04	39.84	37.39	37.77	35.79	36.28
Using POCS as FM’s estimation						
	G		R		B	
Image	FM	POCS	FM	POCS	FM	POCS
1	36.84	32.80	34.34	32.65	34.41	31.78
2	36.33	33.37	32.97	33.38	35.61	33.37
3	37.05	33.19	35.36	33.40	34.30	31.99
4	34.57	31.52	30.47	29.79	30.79	29.26
5	38.40	31.51	34.90	33.69	34.85	31.22
6	39.13	33.47	34.87	34.12	36.48	33.17
7	36.12	32.22	33.74	31.92	32.50	30.57
8	39.67	33.74	37.75	34.81	36.72	33.36
9	35.80	28.12	33.39	30.62	31.94	28.61
10	39.21	31.60	35.99	33.23	35.44	30.91
11	40.19	33.92	36.80	34.16	38.09	35.92
12	39.78	32.50	37.61	33.92	37.58	33.77
13	37.12	32.24	33.54	31.58	35.06	34.01
14	36.31	32.81	31.36	32.81	32.71	32.23
15	38.70	30.71	34.97	31.07	36.26	31.35
16	38.42	28.97	37.16	32.34	35.38	30.99