Minimized-Laplacian Residual Interpolation for Color Image Demosaicking

Daisuke Kiku, Yusuke Monno, Masayuki Tanaka, and Masatoshi Okutomi

Tokyo Institute of Technology

ABSTRACT

A color difference interpolation technique is widely used for color image demosaicking. In this paper, we propose a *minimized-laplacian residual interpolation* (MLRI) as an alternative to the color difference interpolation, where the residuals are the differences between observed and tentatively estimated pixel values. In the MLRI, we estimate the tentative pixel values by minimizing the Laplacian energies of the residuals. This residual image transformation makes the interpolation process more precise than the standard color difference transformation. We incorporate the proposed MLRI into the gradient based threshold free (GBTF) algorithm, which is one of current state-of-the-art Bayer demosaicking algorithms. Experimental results demonstrate that our proposed demosaicking algorithm can outperform the state-of-the-art algorithms for the 30 images of the IMAX and the Kodak datasets.

Keywords: Interpolation, demosaicking, color filter array (CFA), residual, minimized-Laplacian.

1. INTRODUCTION

A single-sensor color imaging technology with a color filter array (CFA) is widely used in a digital camera industry.[1] In a single-sensor camera with the CFA, only one pixel value among RGB values is recorded at each pixel and the other two pixel values are interpolated by a process called demosaicking.[2] Therefore, the development of a high-performance demosaicking algorithm plays a crucial role to acquire high-quality color images.

The most popular and widely used CFA is the Bayer CFA[3] as shown in Fig. 1. Researches on demosaicking algorithms for the Bayer CFA have a long history.[2] Most of the Bayer demosaicking algorithms first interpolate the missing G pixel values since the G pixels have the double sampling density compared with the R and B pixels. Then, the R and B pixel values are transformed into the color difference domains, namely, $\mathrm{R}-\mathrm{G}$ and $\mathrm{B}-\mathrm{G}$, and the interpolations are performed in the color difference domains. Finally, the interpolated G image is added to the interpolated color difference images to acquire full R and B images. We call this sequence of the process the color difference interpolation. It is known that all color bands have very similar image structures such as textures and edges.[4] From this observation, the color difference images can be assumed to be relatively smooth. For this reason, the color difference interpolation yields the high-quality color images than an independent interpolation of each color band.

In this paper, we propose a novel demosaicking algorithm using a *minimized-laplacian residual interpolation* (MLRI). We generate the tentative estimates of the R and B images ($\check{\mathrm{R}}$ and $\check{\mathrm{B}}$) and calculate residuals instead of the color differences. The residuals are the differences between the observed and the tentatively estimated R and B pixel values ($\mathrm{R}-\check{\mathrm{R}}$ and $\mathrm{B}-\check{\mathrm{B}}$). The tentative estimates of the R and B images are generated by minimizing the Laplacian energies of the residuals. The motivation of the Laplacian energy minimization is that the bilinear interpolation can provide better interpolation results for the images with the smaller Laplacian energies. We generate the tentative estimates of the R and B images by upsampling the observed R and B pixel values by using the guided filter,[5] which is a recently-proposed powerful edge-preserving filter. We incorporate the proposed MLRI into the gradient based threshold free (GBTF) algorithm,[6] which is one of state-of-the-art Bayer demosaicking algorithms. Experimental results demonstrate that our proposed demosaicking algorithm using the MLRI can give state-of-the-art performance for the 30 images of the IMAX and the Kodak datasets.

Digital Photography X, edited by Nitin Sampat, Radka Tezaur, et. al. Proc. of SPIE-IS&T
Electronic Imaging, SPIE Vol. 9023, 90230L · © 2014 SPIE-IS&T
CCC code: 0277-786X/14/$18 · doi: 10.1117/12.2038425

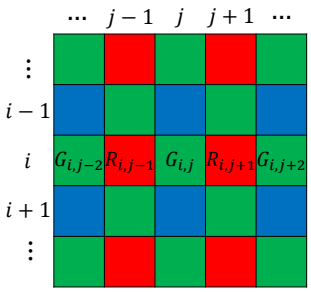


Figure 1. Bayer CFA.

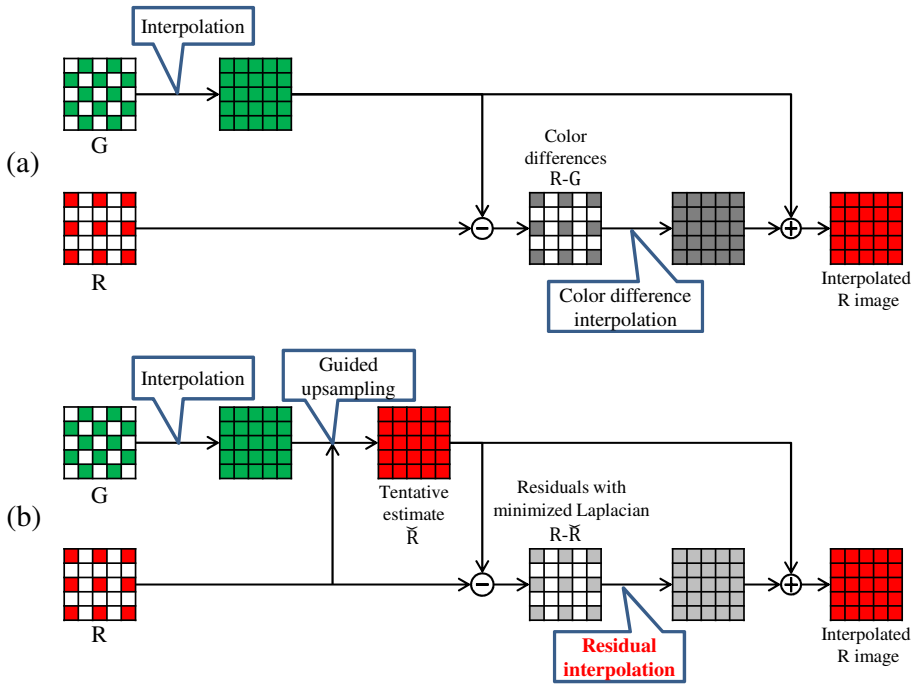


Figure 2. The interpolation of the R pixel values (a) by using the standard color difference interpolation, and (b) by using our proposed MLRI.

2. PROPOSED MINIMIZED-LAPLACIAN RESIDUAL INTERPOLATION

2.1 Outline of the proposed MLRI

We first introduce the basic processing pipeline of the proposed MLRI. Taking the interpolation of the R pixel values as an example, we compare the proposed MLRI with the standard color difference interpolation.

Fig. 2 (a) shows the interpolation process of the R pixel values by using the standard color difference interpolation. First, the G image is interpolated. Then, the color differences (R − G) are calculated at the R pixel locations and the interpolation is performed in the color difference domain. Finally, the G image is added to the interpolated color difference image to acquire the interpolated R image.

Fig. 2 (b) shows the interpolation process of the R pixel values by using the proposed MLRI. First, the G image is interpolated, which is the same as the color difference interpolation. Then, we generate the tentative estimate of the R image by using the guided filter.[5] In this paper, we call this process *guided upsampling*. Next, we calculate the residuals between the observed and the tentatively estimated R pixel values (R − Ř) at the R pixel locations. The tentative estimate of the R image is generate by minimizing the Laplacian energies of the residuals. After that, we interpolate the residuals instead of the color differences. Finally, the tentative estimate of the R image is added to the interpolated residual image to acquire the interpolated R image. In the next section, we describe how we generate the tentative estimate of the R image by the guided upsampling.

2.2 Tentative estimate generation by the guided upsampling

We generate the tentative estimate of the R image by upsampling the observed R pixel values by using the guided filter.[5] The guided filter can accurately upsample input sparse data by using a given guide image, which is used as a reference to exploit image structures.

Fig. 3 shows the outline of the guided upsampling of the R pixel values. For each local window, the guided filter generates the output as the linear transformation of the guide image. We use the interpolated G image as the guide image[7] and generate the tentative estimate Ř in a local window $\omega_{p,q}$ centered at the pixel (p, q) as:

$$\check{R}_{i,j} = a_{p,q} G_{i,j} + b_{p,q}, \quad \forall_{i,j} \in \omega_{p,q}, \tag{1}$$

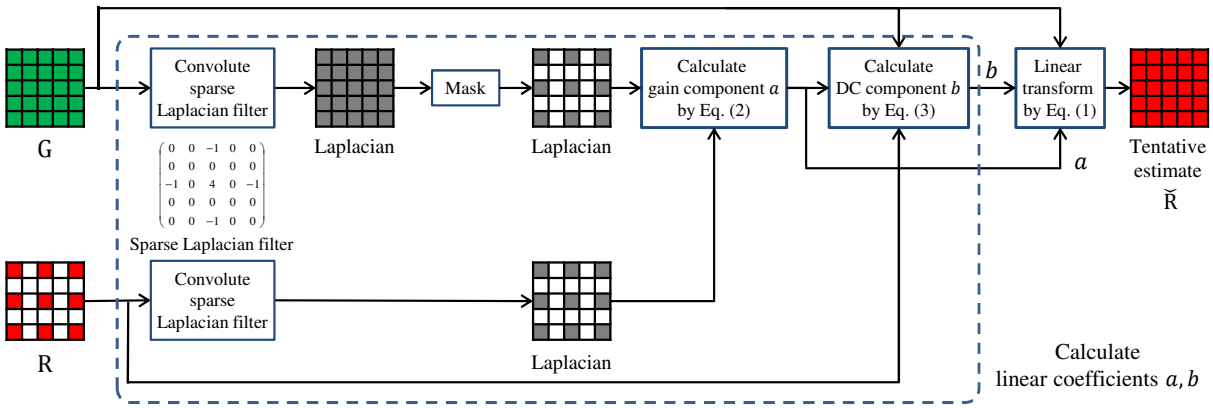


Figure 3. The outline of the guided upsampling of the R pixel values.

where $(a_{p,q}, b_{p,q})$ are linear coefficients assumed to be constant in the local window $\omega_{p,q}$, $a_{p,q}$ is a gain component, $b_{p,q}$ is a DC component, and (i,j) is the pixel in the local window $\omega_{p,q}$. Although the original guided filter minimizes a sum of squared differences, we minimize the Laplacian energies of the residuals in the proposed MLRI. We calculate the gain component $a_{p,q}$ by minimizing the following cost function as:

$$\begin{aligned} a_{p,q} &= \arg\min_{a_{p,q}} \sum_{i,j\in\omega_{p,q}} M_{i,j}\left[\left(\Delta^2\left(R_{i,j}-\check{R}_{i,j}\right)\right)^2\right] \\ &= \arg\min_{a_{p,q}} \sum_{i,j\in\omega_{p,q}} M_{i,j}\left[\left(\Delta^2 R_{i,j} - a_{p,q}\Delta^2 G_{i,j}\right)^2\right], \end{aligned} \tag{2}$$

where $M_{i,j}$ is a binary mask at the pixel (i,j) which is one for the sampled pixels and zero for the others.. Since the observed R pixel values are subsampled, we can not calculate the exact Laplacians of the R image at every pixel. Therefore, we apply a sparse Laplacian filter as shown in Fig. 3 and approximately calculate the Laplacians of the G and R images in a mosaic pattern. Based on the mosaiced Laplacians, we calculate the gain component $a_{p,q}$ by Eq. (2). Although the DC component $b_{p,q}$ can be arbitrary since the DC component does not affect the calculation of the Laplacians, we determine the DC component $b_{p,q}$ as:

$$b_{p,q} = \arg\min_{b_{p,q}} \sum_{i,j\in\omega_{p,q}} M_{i,j}\left(R_{i,j} - a_{p,q}G_{i,j} - b_{p,q}\right)^2. \tag{3}$$

In the above process, the linear coefficients (a,b) are determined in each window, therefore $\check{R}_{i,j}$ in Eq. (1) is not unique when they are calculated in different windows. We simply average these linear coefficients to calculate the final outputs.

3. PROPOSED DEMOSAICKING ALGORITHM

The proposed MLRI can be used as an alternative to the color difference interpolation in an arbitrary demosaicking algorithms which involve the color difference interpolation. In this paper, we develop our proposed demosaicking algorithm by incorporating the MLRI into the GBTF algorithm.[6]

The GBTF algorithm first interpolates the G pixel values, where the color difference interpolation (Hamilton and Adams' interpolation formula[4]) is used. Then, the GBTF algorithm interpolates the R and B pixel values involving the color difference interpolation. We replace the above color difference interpolations with the proposed MLRI for the interpolation of G, R and B pixel values.

3.1 Green interpolation

In this section, we propose the interpolation process of the missing G pixel values by incorporating the MLRI into the GBTF algorithm.

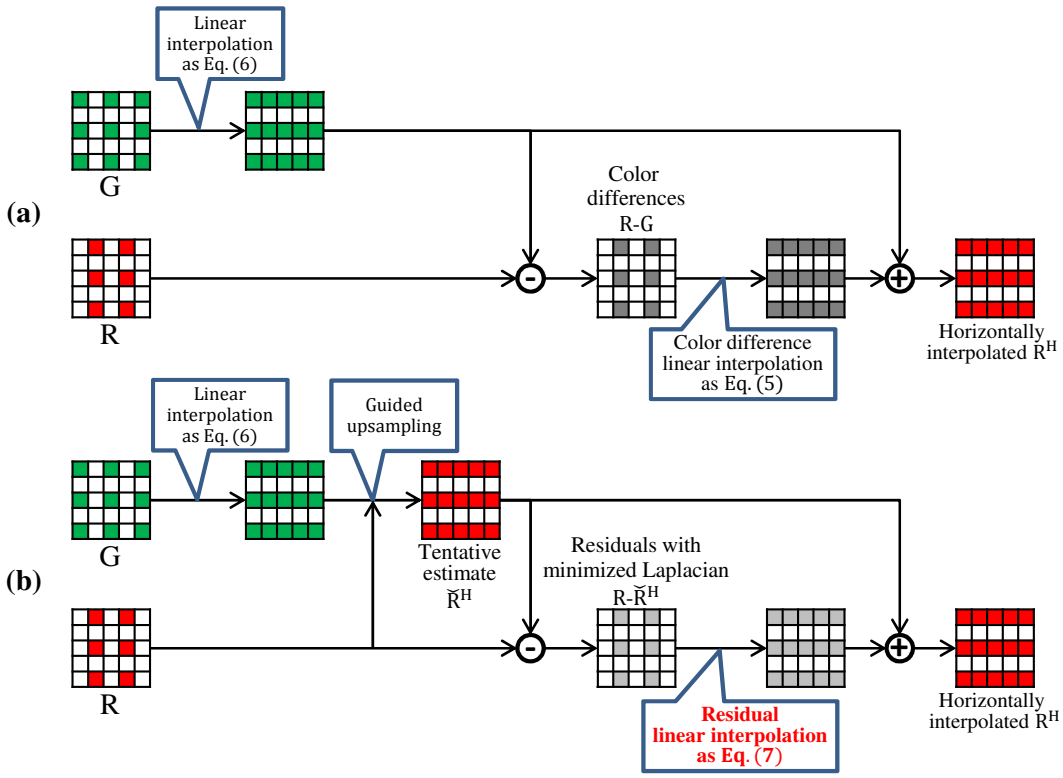


Figure 4. The horizontal R pixel value interpolation in the step (i) (a) by the Hamilton and Adam's interpolation formula, and (b) by the proposed MLRI.

The interpolation process of the G pixel values by the GBTF algorithm consists of three steps: (i) The Hamilton and Adams' interpolation formula[4] is applied in the horizontal and vertical directions to estimate the G pixel values at the R and B pixels and the R or B pixel values at the G pixels. As a result, the horizontally and vertically estimated R, G, and B pixel values are generated. (ii) The horizontal and vertical color differences (G-R or G-B) are calculated at each pixel. Then, the horizontal and vertical color differences are smoothed and combined into the final color difference estimate. (iii) The G pixel values at the R and B pixels are interpolated by adding the observed R or B pixel values to the final color difference estimates.

The Hamilton and Adams' interpolation formula in the step (i) can be interpreted as a linear color difference interpolation as shown in Fig. 4 (a). We replace the linear color difference interpolation with the proposed MLRI as shown in Fig. 4 (b). To simplify the explanation, we focus on the estimation of the R pixel values at the G pixels in the horizontal direction. The B pixel values at the G pixels are estimated in the same manner as the R pixel values. And also, we apply the same process in the vertical direction.

The Hamilton and Adams' interpolation formula in the step (i) for the R pixel value in the horizontal direction can be expressed as:

$$\hat{R}^H_{i,j} = (R_{i,j-1} + R_{i,j+1})/2 + (2\times G_{i,j} - G_{i,j-2} - G_{i,j+2})/4, \tag{4}$$

where the suffix (i, j) represents the target pixel, $\hat{R}^H_{i,j}$ is the horizontally estimated R pixel value at the G pixel. As shown in Fig. (4) (a), this interpolation formula can be interpreted as the linear color difference interpolation as:

$$\hat{R}^H_{i,j} - G_{i,j} = (R_{i,j-1} - \tilde{G}^H_{i,j-1})/2 + (R_{i,j+1} - \tilde{G}^H_{i,j+1})/2, \tag{5}$$

where $\tilde{G}^H$ is the horizontally estimated G pixel value at the R pixel calculated as:

$$\tilde{G}^H_{i,j-1} = (G_{i,j-2} + G_{i,j})/2, \quad \tilde{G}^H_{i,j+1} = (G_{i,j} + G_{i,j+2})/2. \tag{6}$$

In the proposed algorithm, we use the tentative estimates instead of these estimated G pixel values as shown in Fig. 4 (b). First, we apply the linear interpolation to the G pixel values as Eq. (6) to generate the guide. Then, the guided upsampling is applied to obtain the tentative estimates of the R pixel values. After that, the residuals with minimized Laplacian energies are calculated and the residual linear interpolation is performed as:

$$\hat{R}^H_{i,j} - \check{R}^H_{i,j} = (R_{i,j-1} - \check{R}^H_{i,j-1})/2 + (R_{i,j+1} - \check{R}^H_{i,j+1})/2. \tag{7}$$

Finally, we estimate the R pixel values at the G pixels by adding the tentative estimates to the interpolated residuals. The G pixel values at the R pixels are estimated in the same manner, where the tentative estimates of the G pixel values are calculated.

After the above step (i), the step (ii) and step (iii) are performed in the similar manner to the GBTF algorithm. In the step (ii), the color differences for horizontal and vertical directions are calculate as:

$$\begin{aligned}
\tilde{\Delta}^H_{g,r}(i,j) &= \begin{cases} \hat{G}^H_{i,j} - R_{i,j}, & \text{G is interpolated} \\ G_{i,j} - \hat{R}^H_{i,j}. & \text{R is interpolated} \end{cases} \\
\tilde{\Delta}^V_{g,r}(i,j) &= \begin{cases} \hat{G}^V_{i,j} - R_{i,j}, & \text{G is interpolated} \\ G_{i,j} - \hat{R}^V_{i,j}. & \text{R is interpolated} \end{cases}
\end{aligned} \tag{8}$$

The color differences at blue pixels are calculated in the same manner, simply by replacing R with B in the formulas above. Then, the directional color differences are smoothed and combined as:

$$\begin{aligned}
\tilde{\Delta}_{g,r}(i,j) = \{&\omega_N * f * \tilde{\Delta}^V_{g,r}(i-4:i,j) + \\
&\omega_S * f * \tilde{\Delta}^V_{g,r}(i:i+4,j) + \\
&\omega_E * \tilde{\Delta}^H_{g,r}(i,j-4:j) * f^T + \\
&\omega_W * \tilde{\Delta}^H_{g,r}(i,j:j+4) * f^T\}/\omega_T, \\
&\omega_T = \omega_N + \omega_S + \omega_E + \omega_W.
\end{aligned} \tag{9}$$

In the GBTF algorithm, the simple averaging filter, $f = [11111]/5$, is used for smoothing the directional color differences in Eq. (9). In the proposed algorithm, we apply a Gaussian weighted averaging filter, $f = [0.56, 0.35, 0.08, 0.01, 0]$, instead of the simple averaging filter. We empirically use 1 for the standard deviation of the Gaussian weight. This weighted averaging filter improves the performance. The weights for each direction $(\omega_N, \omega_S, \omega_E, \omega_W)$ are calculated using color difference gradients in the horizontal and vertical directions as:

$$\begin{aligned}
\omega_E &= 1/\left(\sum_{a=i-1}^{i+1}\sum_{b=j}^{j+2} D^H_{a,b}\right)^2, \qquad \omega_W = 1/\left(\sum_{a=i-1}^{i+1}\sum_{b=j-2}^{j} D^H_{a,b}\right)^2, \\
\omega_N &= 1/\left(\sum_{a=i-2}^{i}\sum_{b=j-1}^{j+1} D^V_{a,b}\right)^2, \quad \omega_S = 1/\left(\sum_{a=i}^{i+2}\sum_{b=j-1}^{j+1} D^V_{a,b}\right)^2,
\end{aligned} \tag{10}$$

where the directional gradients are calculated as:

$$\begin{aligned}
D^H_{i,j} &= \|\tilde{\Delta}^H_{i,j-1} - \tilde{\Delta}^H_{i,j+1}\|, \\
D^V_{i,j} &= \|\tilde{\Delta}^V_{i-1,j} - \tilde{\Delta}^V_{i+1,j}\|.
\end{aligned} \tag{11}$$

Finally, in the step (iii), we obtain the G pixel values at the R and B pixel locations by adding the observed R or B pixel values as:

$$\begin{aligned}
\tilde{G}(i,j) &= R(i,j) + \tilde{\Delta}_{g,r}(i,j), \\
\tilde{G}(i,j) &= B(i,j) + \tilde{\Delta}_{g,b}(i,j).
\end{aligned} \tag{12}$$

Table 1. The average PSNRs and CPSNRs of the IMAX 18 images, where the bold font represents the best performance.

Algorithms	PSNR			CPSNR
	R	G	B	
AHD[8]	33.00	36.98	32.16	33.49
DLMMSE[9]	34.03	37.99	33.04	34.47
GBTF[6]	33.48	36.59	32.71	33.89
LPA[10]	34.36	37.88	33.30	34.72
LDI-NAT[11]	36.28	39.76	34.39	36.20
RI[12]	36.07	**39.99**	35.35	36.48
Proposed	**36.35**	39.90	**35.36**	**36.62**

Table 2. The average PSNRs and CPSNRs of the Kodak 12 images, where the bold font represents the best performance.

Algorithms	PSNR			CPSNR
	R	G	B	
AHD[8]	38.81	40.84	38.42	39.22
DLMMSE[9]	41.17	43.94	40.51	41.62
GBTF[6]	**41.71**	**44.85**	**41.01**	**42.21**
LPA[10]	41.66	44.46	41.00	42.12
LDI-NAT[11]	38.30	40.49	37.94	38.77
RI[12]	39.64	42.17	38.87	39.99
Proposed	40.59	42.97	39.86	40.94

Table 3. The average PSNRs and CPSNRs of whole IMAX and Kodak 30 images, where the bold font represents the best performance.

Algorithms	PSNR			CPSNR
	R	G	B	
AHD[8]	35.32	38.52	34.66	35.78
DLMMSE[9]	36.89	40.37	36.02	37.33
GBTF[6]	36.77	39.89	36.03	37.22
LPA[10]	37.28	40.51	36.38	37.68
LDI-NAT[11]	37.09	40.05	35.81	37.23
RI[12]	37.50	40.86	36.76	37.88
Proposed	**38.04**	**41.13**	**37.16**	**38.35**

3.2 Red and blue interpolation

After the G image is interpolated, the GBTF algorithm uses the color difference interpolation for the R and B pixel values interpolation as described in Fig. 2 (a). Therefore, we can simply replace the color difference interpolation with the propose MLRI as described in Fig. 2 (b). We use the bilinear interpolation for the residual interpolation.

4. EXPERIMENTS

The proposed algorithm was evaluated with two standard color image datasets, the IMAX dataset and the Kodak dataset[2] *. The IMAX dataset consists of 18 images and the image size is 500×500. The IMAX images are cropped from original 2310×1814 high-resolution images. The Kodak dataset consists of 12 images and the image size is 768×512. We compared the proposed algorithm with six state-of-the-art algorithms; AHD,[8] DLMMSE,[9] GBTF,[6] LPA,[10] LDI-NAT,[11] and RI[12] algorithms. In the RI algorithm, simply residuals, instead of the Laplacian energies of residuals, are minimized in our previous work.

The average PSNRs and CPSNRs of the IMAX 18 images are shown in Table 1. The average CPSNR of the proposed algorithm on the IMAX dataset outperforms all the state-of-the-art algorithms. The average PSNRs and CPSNRs of the Kodak 12 images are shown in Table 2. The average CPSNR of the proposed algorithm on the Kodak dataset is lower than the GBTF, LPA and DLMMSE algorithms. It is remarkable that several algorithms only work well for one dataset, but do not for another dataset. For example, the LDI-NAT algorithm only works well for the IMAX dataset, in contrast, the GBTF algorithm only works well for the Kodak dataset. Then, we evaluated the average PSNRs and CPSNRs of the whole 30 images of the IMAX and the Kodak datasets as shown in Table 3. The proposed algorithm outperforms all the state-of-the-art algorithms in terms

*The source code of our proposed demosaicking algorithm can be downloaded from http://www.ok.ctrl.titech.ac.jp/res/DM/RI.html.

of the total average PSNRs and CPSNR. Fig. 5 shows the visual comparison of the star region in the IMAX dataset, and Fig. 6 shows the visual comparison of the fence region of the lighthouse in the Kodak dataset. From this visual comparisons, we can find that the proposed algorithm can sharply interpolate the images without severe color artifacts.

5. CONCLUSION

In this paper, we have proposed the MLRI for color image demosaicking. The proposed MLRI can be used as an alternative to the widely used color different interpolation. In the proposed MLRI, we perform the residual interpolation, where the residuals are the differences between the observed and the tentatively estimated pixel values. We estimate the tentatively pixel values by minimizing the Laplacian energies of the residuals to make the interpolation process more precise than the standard color difference interpolation. We have also proposed a novel demosaicking algorithm by incorporating the proposed MLRI into the GBTF algorithm. Experimental results demonstrate that our proposed demosaicking algorithm can outperform the state-of-the-art algorithms for the 30 images of the IMAX and the Kodak datasets.

REFERENCES

[1] Lukac, R., [*Single-sensor imaging: methods and applications for digital cameras*], CRC Press (2008).

[2] Li, X., Gunturk, B., and Zhang, L., "Image demosaicing: a systematic survey," *Proc. of SPIE* **6822**, 68221J–68221J–15 (2008).

[3] Bayer, B., "Color imaging array," *U.S. Patent 3971065* (1976).

[4] Hamilton Jr, J. and Adams Jr, J., "Adaptive color plan interpolation in single sensor color electronic camera," (1997). US Patent 5,629,734.

[5] He, K., Sun, J., and Tang, X., "Guided image filtering," *Proc. of the 11th European Conf. on Computer Vision (ECCV)* **6311**, 1–14 (2010).

[6] Pekkucuksen, I. and Altunbasak, Y., "Gradient based threshold free color filter array interpolation," *Proc. of IEEE Int. Conf. on Image Processing (ICIP)* , 137–140 (2010).

[7] Monno, Y., Tanaka, M., and Okutomi, M., "Multispectral demosaicking using guided filter," *Proc. of SPIE* **8299**, 82990O–1–82990O–7 (2012).

[8] Hirakawa, K. and Parks, T. W., "Adaptive homogeneity-directed demosaicking algorithm," *IEEE Trans. on Image Processing* **14**(3), 360–369 (2005).

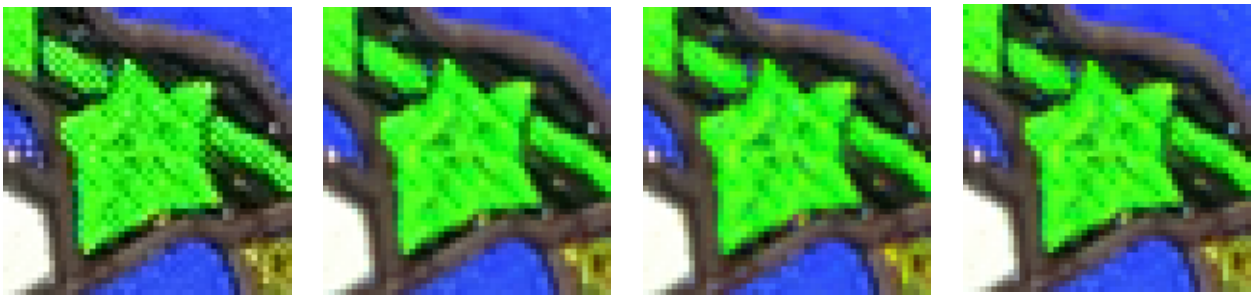
[9] Zhang, L. and Wu, X., "Color demosaicking via directional linear minimum mean square-error estimation," *Image Processing, IEEE Transactions on* **14**(12), 2167–2178 (2005).

[10] Paliy, D., Katkovnik, V., Bilcu, R., Alenius, S., and Egiazarian, K., "Spatially adaptive color filter array interpolation for noiseless and noisy data," *Int. Journal of Imaging Systems and Technology* **17**(3), 105–122 (2007).

[11] Zhang, L., Wu, X., Buades, A., and Li, X., "Color demosaicking by local directional interpolation and nonlocal adaptive thresholding," *Journal of Electronic Imaging* **20**(2), 023016–023016 (2011).

[12] Kiku, D., Monno, Y., Tanaka, M., and Okutomi, M., "Residual interpolation for color image demosaicking," *Proc. of IEEE Int. Conf. on Image Processing (ICIP)* , 2304–2308 (2013).

Original DLMMSE[9] GBTF[6]



LPA[10] LDI-NAT[11] RI[12] Proposed

Figure 5. Visual comparison for the star region in the IMAX dataset.

Lighthouse Original DLMMSE[9] GBTF[6]

LPA[10] LDI-NAT[11] RI[12] Proposed

Figure 6. Visual comparison for the fence region of the lighthouse in the Kodak dataset.