

Beyond Color Difference: Residual Interpolation for Color Image Demosaicking

Daisuke Kiku, Yusuke Monno, *Member, IEEE*, Masayuki Tanaka, *Member, IEEE*,
and Masatoshi Okutomi, *Member, IEEE*

Abstract—In this paper, we propose residual interpolation (RI) as an alternative to color difference interpolation, which is a widely accepted technique for color image demosaicking. Our proposed RI performs the interpolation in a residual domain, where the residuals are differences between observed and tentatively estimated pixel values. Our hypothesis for the RI is that if image interpolation is performed in a domain with a smaller Laplacian energy, its accuracy is improved. Based on the hypothesis, we estimate the tentative pixel values to minimize the Laplacian energy of the residuals. We incorporate the RI into the gradient-based threshold free algorithm, which is one of the state-of-the-art Bayer demosaicking algorithms. Experimental results demonstrate that our proposed demosaicking algorithm using the RI surpasses the state-of-the-art algorithms for the Kodak, the IMAX, and the beyond Kodak data sets.

Index Terms—Bayer color filter array, demosaicking, residual interpolation, guided upsampling.

I. INTRODUCTION

SINGLE-SENSOR color imaging using a color filter array (CFA) is widely used in the current 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 must be generated by interpolation. This interpolation process, which is typically called a demosaicking process, plays a crucial role in acquiring high-quality color images.

The most popular and widely used CFA is the Bayer CFA (Fig. 1) [2]. Demosaicking algorithms for the Bayer CFA have extensively been studied [3]–[5]. Most Bayer demosaicking algorithms interpolate the missing G pixel values first, because the G pixels have a double sampling density of the R and the B pixels. For the interpolation of the missing R and B pixel values, the observed R and B pixel values are generally transformed into color ratio [6], [7] or color difference domains [8]–[10]. Then, interpolation

Manuscript received October 7, 2014; revised March 6, 2015 and July 7, 2015; accepted December 30, 2015. Date of publication January 14, 2016; date of current version February 2, 2016. This work was supported by the Strategic Information and Communications Research and Development Promotion Programme through the Ministry of Internal Affairs and Communications under Grant 141203024. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Qionghai Dai.

The authors are with the Department of Mechanical and Control Engineering, Tokyo Institute of Technology, Tokyo 152-8550, Japan (e-mail: dkiku@ok.ctrl.titech.ac.jp; ymonno@ok.ctrl.titech.ac.jp; mtanaka@ctrl.titech.ac.jp; mxo@ctrl.titech.ac.jp).

Color versions of one or more of the figures in this paper are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/TIP.2016.2518082

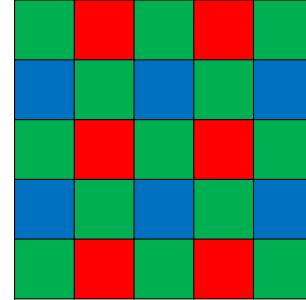


Fig. 1. The Bayer CFA.

is performed in the transformed domains. Another popular approach is a frequency domain algorithm [11]–[14], which transforms the CFA image into the frequency domain, then separates luminance and chrominance components by frequency filtering. The algorithms based on a theory of compressed sensing have also been proposed [15], [16].

In the past literatures, many algorithms have been proposed based on the color difference interpolation because of its simplicity of implementation. The assumption for the color difference interpolation is that all color bands have spectral correlations and similar image structures such as textures and edges. Therefore, high-frequency energies are reduced in the color difference domains (i.e., $R - G$ and $B - G$), which simplifies the interpolation process. For this reason, the color difference interpolation improves the demosaicking accuracy compared with the independent interpolation of each color band.

Recent sophisticated algorithms using the color difference interpolation are different regarding directional interpolation used for generating the G image. Some algorithms decide the direction a posteriori based on homogeneity metrics on LAB space [17], gradients of color differences [18]–[20], and variance of color differences [21]–[23]. An iterative approach [24], [25] and a nonlocal approach [26]–[28] have also been proposed. We refer to the survey papers [3]–[5] for more details because it is too diverse to explain all existing algorithms here.

In this paper, we propose residual interpolation (RI) as an alternative to the color difference interpolation. The RI performs the interpolation in a “residual” domain. Instead of calculating the standard color differences, we generate tentative estimates of the R and the B images (\tilde{R} and \tilde{B}) and calculate their residuals, which are the differences between

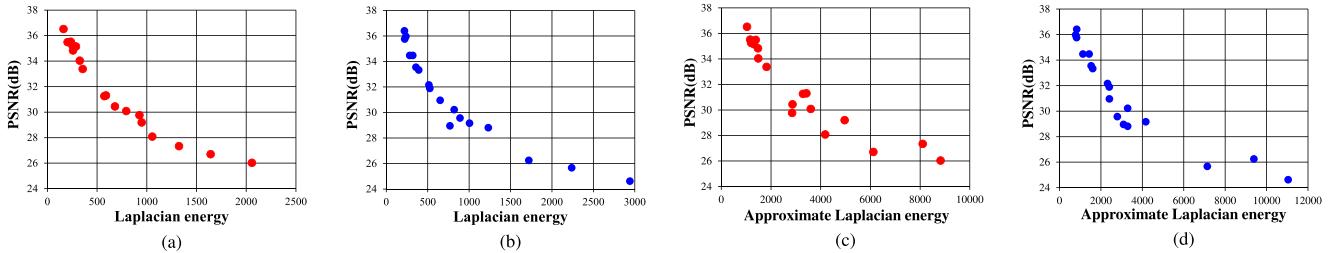


Fig. 2. (a), (b) Laplacian energy vs. PSNR plots, and (c), (d) approximate Laplacian energy vs. PSNR plots for the R and B images of the IMAX 18 images.

the observed and the tentatively estimated pixel values (i.e., $R - \hat{R}$ and $B - \hat{B}$) [29], [30]. The original RI, which is our initial work in [29], generates the tentative estimates by minimizing the residuals themselves using the guided filtering (GF) [31]. Instead, its extended version called minimized-Laplacian residual interpolation (MLRI) [30] generates the tentative estimates by minimizing the Laplacian energy of the residuals. Our hypothesis for the MLRI is that if image interpolation is performed in a domain with a smaller Laplacian energy, its accuracy is improved. For example, the MLRI makes the interpolation process more precise because successive bilinear interpolation can provide better interpolation results for the images with smaller Laplacian energies. The MLRI can generally be incorporated with arbitrary demosaicking algorithms that involve the color difference interpolation. We incorporate the MLRI into the gradient based threshold free (GBTF) algorithm [19], which is one of the state-of-the-art Bayer demosaicking algorithms. Experimental results demonstrate that our proposed demosaicking algorithm using the MLRI outperforms existing state-of-the-art algorithms for the Kodak [32]–[34], the IMAX [27], and the beyond Kodak [35] datasets.

This paper is the extended and the detailed version of our previous papers [29], [30]. The novel parts of this paper are as follows: (i) we describe statistical observations for the interpolation accuracy with respect to the Laplacian energy in Section II, (ii) we improve the demosaicking accuracy by introducing an weighted averaging for the guided filtering in Section III, (iii) we report a new experimental comparison of the standard color difference interpolation and the proposed MLRI in Section V-A, and (iv) we report additional results on various high-resolution color image datasets including the recently published beyond Kodak dataset in Section V-B.

The rest of this paper is organized as follows. Section II presents our basic observation for the interpolation accuracy with respect to the Laplacian energy. Section III explains the algorithm of our proposed MLRI. Section IV describes the proposed demosaicking algorithm using the MLRI. Section V presents experimental results. Finally, Section VI concludes the paper.

II. BASIC OBSERVATION FOR INTERPOLATION ACCURACY WITH RESPECT TO LAPLACIAN ENERGY

We first examine the relationship between the Laplacian energy and the interpolation performance to recover an original

image from regularly subsampled data, which is our basic observation for the proposed MLRI. The Laplacian energy is a metric used to evaluate the smoothness of images. The Laplacian map $\Delta\mathbf{I}$ of an image \mathbf{I} is calculated as

$$\Delta\mathbf{I} = \begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix} \otimes \mathbf{I}, \quad (1)$$

where \otimes is a convolution operation. The Laplacian energy of the image $E(\mathbf{I})$ is calculated as

$$E(\mathbf{I}) = \frac{1}{|\Omega|} \sum_{i,j \in \Omega} (\Delta I_{i,j})^2, \quad (2)$$

where the suffix (i, j) is a pixel index, Ω is a set of whole image pixel indexes, and $|\Omega|$ denotes the number of whole image pixels. Generally speaking, it is expected that the interpolation performance of a smooth image with a smaller Laplacian energy is better. In fact, bilinear interpolation can perfectly recover images that have zero Laplacian energy. We experimentally analyze this relationship using the IMAX 18 images [27]. In the analysis, we subsample original full R and B images by a scale factor of two so as to be the same sampling pattern as the Bayer CFA. Then, we interpolate the subsampled data by bilinear interpolation. Fig. 2 (a) and Fig. 2 (b) show Laplacian energy vs. PSNR performance plots for the R and the B images of the IMAX 18 images. From these plots, we can infer that a smaller Laplacian energy of the image gives better interpolation performance.

In the above analysis, we show that interpolation performance can generally be improved if performed in a domain with smaller Laplacian energies. However, in the actual demosaicking process, we cannot calculate the exact Laplacian energy from the subsampled R and B images. Therefore, we introduce a sparse Laplacian filter and use an approximate Laplacian energy for the smoothness metric. Here, we consider a subsampled image that has the same sampling pattern as the R or the B band of the Bayer CFA. The approximate Laplacian map $\tilde{\Delta}\mathbf{I}$ of the subsampled image \mathbf{I} is calculated as

$$\tilde{\Delta}\mathbf{I} = \begin{bmatrix} 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & 4 & 0 & -1 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 \end{bmatrix} \otimes \mathbf{I}. \quad (3)$$

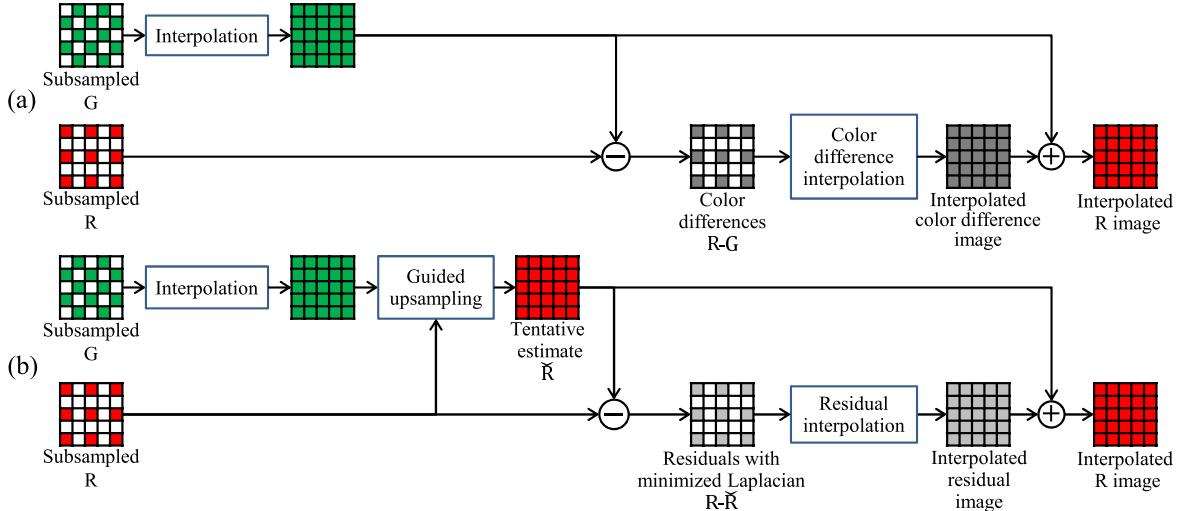


Fig. 3. Outline of the interpolation process of R pixel values (a) using the standard color difference interpolation and (b) using our proposed MLRI.

The approximate Laplacian energy of the subsampled image $\tilde{E}(\mathbf{I})$ is calculated as

$$\tilde{E}(\mathbf{I}) = \frac{1}{|\omega|} \sum_{i,j \in \omega} (\tilde{\Delta} I_{i,j})^2, \quad (4)$$

where ω is a set of subsampled pixel indexes and $|\omega|$ denotes the number of the subsampled pixels.

We conduct the same analysis using the approximate Laplacian energy. Fig. 2 (c) and Fig. 2 (d) show the approximate Laplacian energy vs. PSNR performance plots for the R and the B images of the IMAX 18 images. From these plots, a smaller approximate Laplacian energy also gives better interpolation performance. This observation is used for our proposed MLRI in the next section.

III. PROPOSED MINIMIZED-LAPLACIAN RESIDUAL INTERPOLATION

A. Outline

We describe a basic processing pipeline of the proposed MLRI by taking the interpolation of the R pixel values as an example. Fig. 3 (a) outlines the interpolation process of the R pixel values using standard color difference interpolation. First, the G image is generated by an arbitrary interpolation algorithm. Second, the color differences ($R - G$) are calculated for the R pixel locations. Third, color difference interpolation is performed. Finally, the interpolated G image is added to the interpolated color difference image to obtain the interpolated R image.

Fig. 3 (b) outlines the interpolation process of the R pixel values by the proposed MLRI. First, the G image is generated by an arbitrary interpolation algorithm, which is the same as the color difference interpolation. Second, we generate the tentative estimate of the R image (\check{R}) and calculate the residuals between the observed and the tentatively estimated R pixel values ($R - \check{R}$). The tentative estimate is generated from the interpolated G image by guided upsampling, where the approximate Laplacian energy of the subsampled residuals

is minimized. Third, we interpolate the residuals instead of the color differences. Finally, the tentative estimate is added to the interpolated residual image to obtain the interpolated R image.

B. Tentative Estimate Generation by Guided Upsampling

In the proposed MLRI, we generate the tentative estimate by guided upsampling using the GF [31]. The GF is a powerful edge-preserving filtering and can be used as an alternative to well-known bilateral filtering [36]. One advantage of the GF over the bilateral filtering is that the computing time is independent of the filter size. The usefulness of the GF is verified in many applications such as image matting, haze removal, detail enhancement, and image upsampling [31]. Because of its effectiveness, the GF is widely spreading to the computer vision and image processing communities and recently included in official MATLAB and OpenCV functions.

We use the GF for the image upsampling operation to generate the tentative estimates. The GF can accurately upsample input sparse data with an effective guide image, which is used as a reference to exploit image structures. For each local window, the GF generates the output as a linear transformation of the guide image. We use the interpolated G image as the guide image (as in [37] and [38]) and generate the tentative estimate of the R image in a local window $\omega_{p,q}$ centered at the pixel (p, q) as

$$\check{R}_{i,j} = a_{p,q} G_{i,j} + b_{p,q}, \quad \forall i,j \in \omega_{p,q}, \quad (5)$$

where $(a_{p,q}, b_{p,q})$ are the linear coefficients assumed to be constant in the window $\omega_{p,q}$, $a_{p,q}$ is a gain component, $b_{p,q}$ is a DC component, and (i, j) is a pixel index in the window $\omega_{p,q}$. In our experiments, we empirically use the 5×5 window for the R and the B interpolation in Section IV-B, while we use the 3×3 window for the G interpolation in Section IV-A. Although the original GF, which is used in the original RI [29], minimizes a sum of squared differences between the R image and the tentative estimate, i.e., $(R_{i,j} - \check{R}_{i,j})^2$, the MLRI minimizes the approximate Laplacian energy of the residuals as described below.

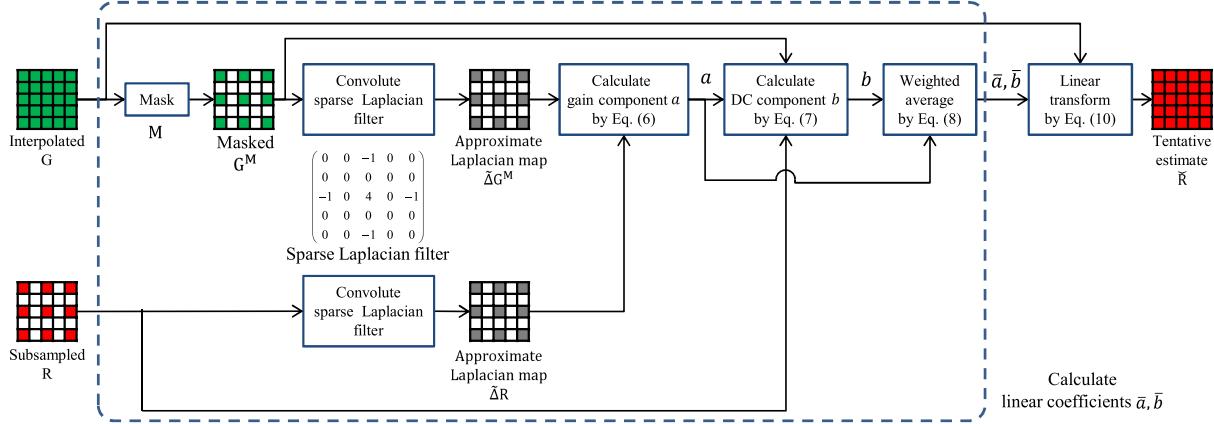


Fig. 4. Outline of the tentative estimate generation by guided upsampling.

Fig. 4 outlines the tentative estimate generation of the R image by the guided upsampling of the observed R pixel values. We first convolute the sparse Laplacian filter to calculate the approximate Laplacian map of the masked G image and the observed R image. Then, for each local window, we calculate the gain component $a_{p,q}$ by minimizing the following cost function.

$$\begin{aligned} E(a_{p,q}) &= \sum_{i,j \in \omega_{p,q}} \left(M_{i,j} \tilde{\Delta} (R_{i,j} - \check{R}_{i,j}) \right)^2, \\ &= \sum_{i,j \in \omega_{p,q}} \left(M_{i,j} \tilde{\Delta} (R_{i,j} - a_{p,q} G_{i,j} - b_{p,q}) \right)^2, \\ &= \sum_{i,j \in \omega_{p,q}} \left(\tilde{\Delta} R_{i,j} - a_{p,q} \tilde{\Delta} (G_{i,j}^M) \right)^2, \quad (6) \end{aligned}$$

where $M_{i,j}$ is a binary mask at the pixel (i, j) , which is one for the sampled R pixels and zero for the others, and $G_{i,j}^M$ denotes the masked G pixel value by $M_{i,j}$. Although the DC component $b_{p,q}$ can be arbitrary because the DC component does not affect the calculation of the Laplacian energy, we determine the DC component $b_{p,q}$ by minimizing the following cost function that minimizes the residual energy given $a_{p,q}$.

$$E(b_{p,q}) = \sum_{i,j \in \omega_{p,q}} \left(R_{i,j} - a_{p,q} G_{i,j}^M - b_{p,q} \right)^2. \quad (7)$$

In the above processes, the linear coefficients $(a_{p,q}, b_{p,q})$ are determined in each local window. Therefore, the resulting $\check{R}_{i,j}$ values are not unique when they are calculated in different windows. We introduce an weighted average of these linear coefficients $(\bar{a}_{i,j}, \bar{b}_{i,j})$, which is an improvement from [30], to calculate the final output as

$$\check{R}_{i,j} = \bar{a}_{i,j} G_{i,j} + \bar{b}_{i,j}, \quad (8)$$

where

$$\begin{aligned} \bar{a}_{i,j} &= \frac{\sum_{p,q \in \omega_{i,j}} W_{p,q} a_{p,q}}{\sum_{p,q \in \omega_{i,j}} W_{p,q}}, \\ \bar{b}_{i,j} &= \frac{\sum_{p,q \in \omega_{i,j}} W_{p,q} b_{p,q}}{\sum_{p,q \in \omega_{i,j}} W_{p,q}}. \quad (9) \end{aligned}$$

The weight $W_{p,q}$ is calculated based on the residual cost as

$$W_{p,q} = 1 / |\omega_{p,q}| \sum_{i,j \in \omega_{p,q}} \left(R_{i,j} - a_{p,q} G_{i,j}^M - b_{p,q} \right)^2, \quad (10)$$

where $|\omega_{p,q}|$ is the number of the sampled R pixels within the window $\omega_{p,q}$.

IV. PROPOSED DEMOSAICKING ALGORITHM

We incorporate the proposed MLRI into the GBTF algorithm [19], which is one of current state-of-the-art Bayer demosaicking algorithms.

A. Green Interpolation

The GBTF algorithm interpolates the missing G pixel values first. The interpolation process of the G pixel values consists of four steps: (i) The Hamilton and Adams' (HA) interpolation formula [10] is applied in the horizontal and vertical directions to estimate the G pixel values at the R and B pixels and the R and B pixel values at the G pixels. As a result, the horizontally and vertically interpolated R, G, and B pixel values are generated. (ii) The horizontal and vertical color differences ($R - G$ and $B - G$) are calculated for each pixel. (iii) The horizontal and vertical color differences are smoothed and then combined into the final color difference estimate. (iv) 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.

Fig. 5 outlines the proposed G pixel value interpolation at the R pixels. The G pixel value interpolation at the B pixels is performed in the same manner. The HA interpolation formula [10] in the step (i) of the GBTF algorithm can be interpreted as horizontal and vertical linear color difference interpolation. We replace the color difference interpolation with the proposed MLRI.

Here, we only consider the horizontal interpolation of the subsampled R pixel values. The vertical interpolation and the interpolation of the subsampled G pixel values are performed in the same manner. The horizontally interpolated

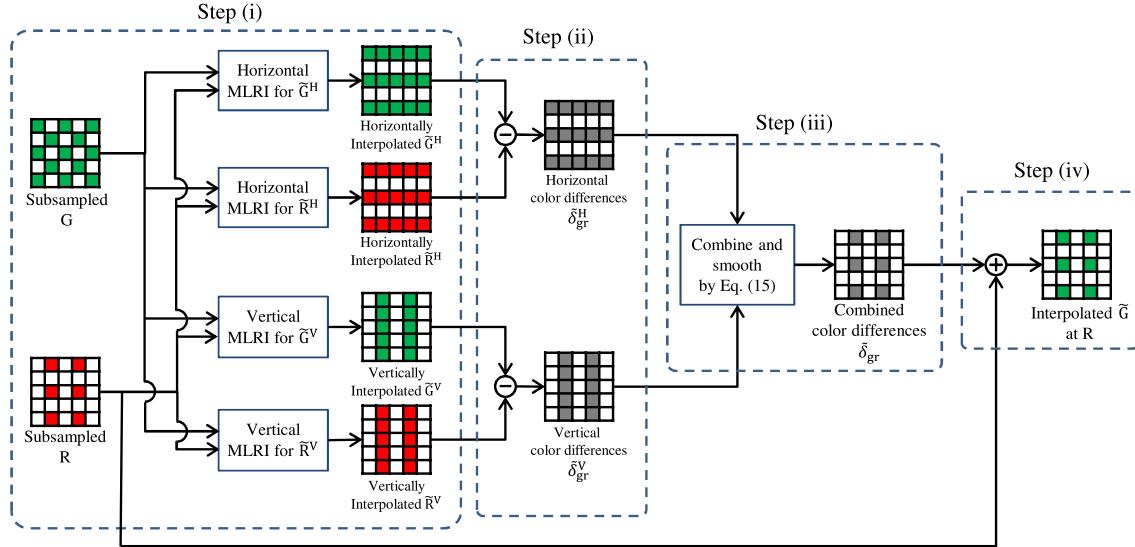


Fig. 5. Outline of the proposed G pixel value interpolation at the R pixels.

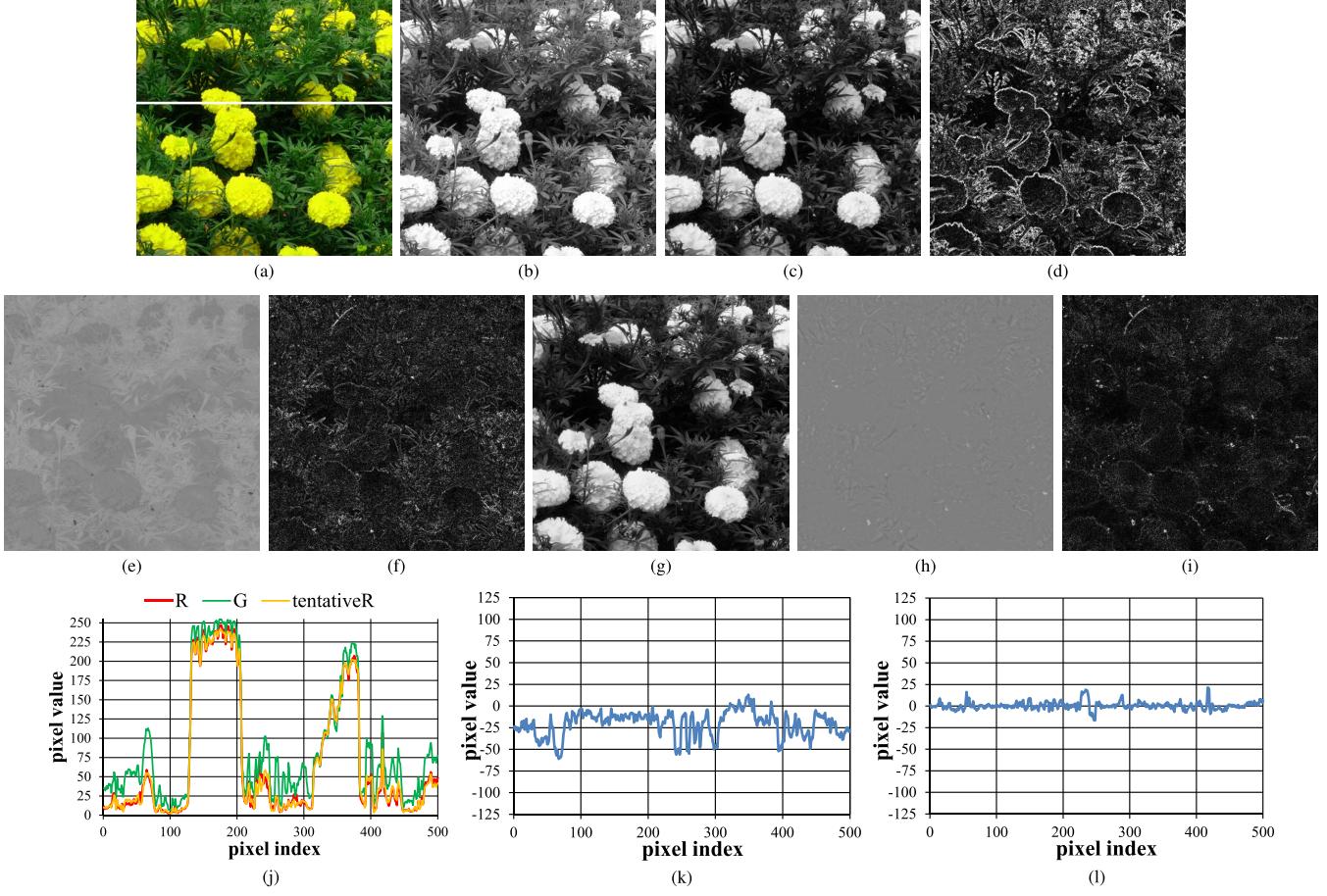


Fig. 6. Examples of (a) RGB image, (b) G image, (c) R image, (d) Laplacian map of the R image, (e) color difference image, (f) Laplacian map of the color difference image, (g) Tentative estimate of the R image, (h) Residual image, and (i) Laplacian map of the residual image. (j) The sliced plots of the original R, the original G, and the tentative estimate images, (k) the sliced plot of the color difference image, and (l) the sliced plot of the residual image.

R pixel value \tilde{R}^H by the HA interpolation formula can be expressed as

$$\tilde{R}_{i,j}^H = (R_{i,j-1} + R_{i,j+1})/2 + (2*G_{i,j} - G_{i,j-2} - G_{i,j+2})/4, \quad (11)$$

where the suffix (i, j) represents a target pixel. The HA interpolation formula can be interpreted as the horizontal linear color difference interpolation as follows:

$$\tilde{R}_{i,j}^H - G_{i,j} = (R_{i,j-1} - \hat{G}_{i,j-1}^H)/2 + (R_{i,j+1} - \hat{G}_{i,j+1}^H)/2, \quad (12)$$

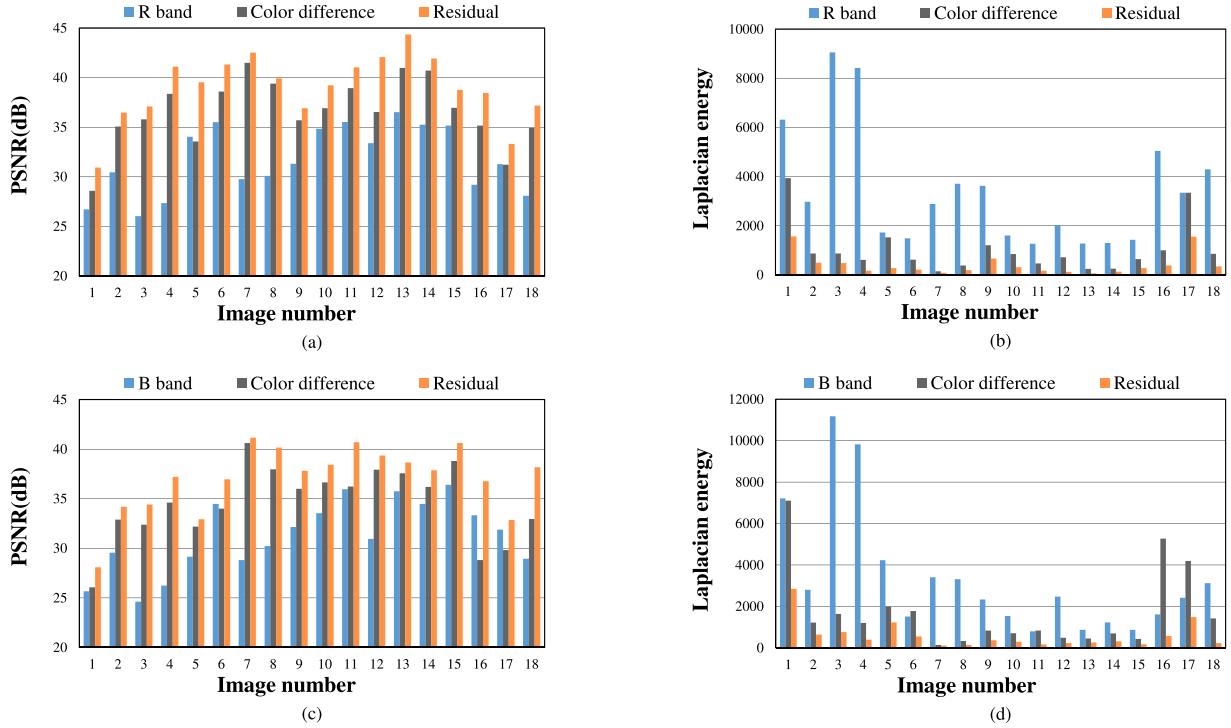


Fig. 7. PSNR performances and approximate Laplacian energies of the original R and B images, the color difference images, and the residual images for the IMAX 18 images. (a) PSNR performances for the R, the color difference, and the residual images. (b) Approx. Laplacian energies of the R, the color difference, and the residual images. (c) PSNR performances for the B, the color difference, and the residual images. (d) Approx. Laplacian energies of the B, the color difference, and the residual images.

TABLE I
STATISTICS OF THE DATASETS

Datasets		IMAX [27]	Low-resolution Kodak [32]	High-resolution Kodak [33], [34]	ARRI [35]
Mean Spectral Correlation	G and R	0.7445	0.8722	0.7866	0.7784
	G and B	0.7114	0.9056	0.9083	0.9473
Mean Saturation		26.45	15.6188	17.1618	12.0921
Mean Chromatic Gradient		4.6766	1.7870	1.9301	4.5453

TABLE II
CPSNR PERFORMANCE FOR THE IMAX 18 IMAGES, WHERE BOLD TYPEFACE REPRESENTS THE BEST PERFORMANCE

No.	AP	SA	AHD	DLM	LPA	DFPD	GBTF	LDI-NAT	LSLC	Proposed		
										RI	MLRI	MLRI+wei.
1	25.59	23.82	26.15	26.98	26.81	26.86	26.42	29.18	25.04	29.12	28.97	29.41
2	32.46	32.14	32.99	33.70	33.85	33.63	33.23	35.05	32.10	35.02	35.10	35.35
3	31.63	31.48	31.27	32.60	32.48	32.12	32.32	32.61	31.73	33.77	33.88	34.05
4	33.23	33.78	33.50	34.34	34.95	34.04	34.22	35.96	34.11	37.89	37.67	38.00
5	29.98	29.33	30.57	31.28	31.40	31.17	30.28	34.10	29.23	33.94	34.02	34.43
6	31.98	30.92	32.72	33.84	34.36	33.81	32.34	37.86	31.14	38.32	38.29	38.83
7	37.82	37.70	36.02	38.68	39.22	37.25	39.16	35.98	38.62	36.98	37.49	37.04
8	36.62	36.98	36.05	37.48	37.89	37.66	37.05	37.52	36.25	36.99	36.97	37.30
9	33.28	32.53	33.64	34.42	35.07	34.46	33.67	37.02	32.50	35.94	36.46	36.84
10	34.97	35.32	35.62	36.36	37.03	36.42	35.71	38.81	34.26	38.18	38.66	39.12
11	35.97	36.18	36.28	37.26	37.71	37.05	36.49	39.60	35.20	39.46	39.95	40.21
12	35.78	36.15	36.01	36.63	36.98	36.53	36.17	38.91	35.13	39.64	39.68	39.84
13	37.47	38.06	38.21	38.82	39.30	38.58	38.19	40.79	36.86	40.32	40.56	40.66
14	36.25	36.63	36.78	37.26	37.54	37.08	36.72	38.73	35.66	38.95	38.79	39.11
15	36.35	36.60	36.80	37.29	37.67	37.06	36.65	38.99	35.70	38.40	38.94	39.25
16	29.02	27.65	29.35	30.46	29.47	30.15	29.44	33.85	28.23	35.17	35.09	35.42
17	27.99	26.01	28.16	29.32	29.27	29.35	28.31	32.88	27.19	32.45	32.59	33.19
18	32.49	32.39	32.76	33.90	33.93	33.71	33.73	35.11	32.30	36.52	36.12	36.41
Ave.	33.27	32.98	33.49	34.48	34.72	34.27	33.89	36.27	32.85	36.50	36.62	36.91

where \hat{G}^H is the linearly interpolated G pixel value as

$$\begin{aligned}\hat{G}_{i,j-1}^H &= (G_{i,j-2} + G_{i,j})/2, \\ \hat{G}_{i,j+1}^H &= (G_{i,j} + G_{i,j+2})/2.\end{aligned}\quad (13)$$

We replace the horizontal linear color difference interpolation with the horizontal linear MLRI. The horizontal linear MLRI is performed in the same process flow as non-directional MLRI (Fig. 3 and Fig. 4). First, we linearly interpolate the

TABLE III
S-CIELAB PERFORMANCE FOR THE IMAX 18 IMAGES, WHERE BOLD TYPEFACE REPRESENTS THE BEST PERFORMANCE

No.	AP	SA	AHD	DLM	LPA	DFPD	GBTF	LDI-NAT	LSLC	Proposed		
										RI	MLRI	MLRI+wei.
1	3.958	5.442	3.856	3.186	3.393	3.204	3.474	2.500	4.242	2.481	2.511	2.353
2	1.487	1.714	1.548	1.317	1.269	1.273	1.365	1.104	1.573	1.027	1.044	0.995
3	2.439	2.515	2.367	2.066	2.093	2.187	2.048	1.848	2.169	1.589	1.616	1.581
4	1.728	1.743	1.655	1.429	1.159	1.375	1.326	0.993	1.423	0.657	0.775	0.698
5	1.904	2.169	1.874	1.620	1.528	1.624	1.728	1.240	2.089	1.171	1.176	1.105
6	1.883	2.379	1.951	1.519	1.414	1.457	1.818	0.995	2.061	0.917	0.930	0.892
7	1.076	1.053	1.060	0.939	0.877	1.119	0.877	1.177	0.970	1.119	1.075	1.115
8	0.758	0.725	0.733	0.631	0.584	0.631	0.661	0.588	0.720	0.578	0.572	0.565
9	1.560	1.809	1.622	1.390	1.217	1.382	1.401	0.999	1.668	1.002	1.028	0.943
10	1.226	1.262	1.328	1.103	0.988	1.076	1.164	0.878	1.358	0.863	0.822	0.774
11	0.953	0.995	0.994	0.818	0.774	0.822	0.906	0.635	1.043	0.648	0.637	0.605
12	1.168	1.214	1.313	1.132	1.100	1.100	1.205	0.880	1.288	0.798	0.782	0.782
13	0.901	0.883	0.934	0.782	0.742	0.799	0.800	0.697	0.949	0.669	0.666	0.652
14	0.893	0.899	0.947	0.818	0.779	0.850	0.861	0.721	0.974	0.684	0.701	0.679
15	0.928	0.950	1.048	0.881	0.832	0.898	0.932	0.750	1.013	0.764	0.743	0.712
16	2.838	3.636	2.577	2.074	2.600	2.075	2.619	1.408	3.106	1.377	1.396	1.341
17	2.951	4.437	3.603	2.810	2.742	2.616	3.021	1.830	3.510	1.791	1.821	1.711
18	1.785	1.931	1.716	1.519	1.520	1.487	1.534	1.254	1.871	1.080	1.186	1.125
Ave.	1.691	1.986	1.729	1.446	1.423	1.443	1.541	1.139	1.779	1.068	1.082	1.035

TABLE IV
CPSNR PERFORMANCE OF THE HIGH-RESOLUTION KODAK 24 IMAGES, WHERE BOLD TYPEFACE REPRESENTS THE BEST PERFORMANCE

No.	AP	SA	AHD	DLM	LPA	DFPD	GBTF	LDI-NAT	LSLC	Proposed		
										RI	MLRI	MLRI+wei.
1	41.70	42.01	41.16	42.70	42.83	42.30	42.30	42.38	41.36	43.40	43.43	43.41
2	35.67	35.76	34.94	36.42	36.35	36.10	35.91	38.25	35.37	40.06	40.23	40.44
3	43.49	43.48	42.63	44.11	44.28	43.43	43.73	44.37	42.82	44.72	45.11	45.02
4	40.73	40.58	39.68	41.33	41.66	41.13	40.43	43.20	40.11	43.96	44.26	44.38
5	42.06	41.84	41.49	42.83	43.30	42.49	41.89	43.63	40.98	43.93	44.22	44.25
6	43.10	43.36	42.66	44.01	44.21	43.56	43.46	44.05	42.55	44.00	44.52	44.54
7	43.28	43.14	42.17	43.81	43.98	43.02	43.39	43.86	42.67	44.03	44.33	44.24
8	36.95	37.10	35.76	37.43	37.41	36.94	37.30	36.95	36.94	37.52	37.67	37.54
9	40.37	40.27	38.68	40.67	40.68	39.69	40.65	39.77	40.55	39.52	40.12	39.75
10	40.76	40.64	39.13	41.07	41.04	40.14	40.89	40.30	40.88	40.13	40.62	40.31
11	41.57	41.77	41.10	42.53	42.77	42.04	41.90	42.92	40.95	42.95	43.34	43.34
12	46.11	46.21	45.26	46.83	46.94	46.08	46.57	46.49	45.85	46.29	46.92	46.81
13	38.31	38.47	37.46	39.03	38.64	38.85	38.07	39.65	37.56	40.26	40.43	40.46
14	40.94	40.82	40.80	42.00	42.65	41.65	41.43	43.09	39.65	43.18	43.80	43.89
15	43.12	43.00	42.42	43.92	44.37	43.59	42.97	45.07	42.42	44.88	45.52	45.53
16	43.13	43.49	42.68	43.97	44.20	43.71	43.66	43.95	42.63	44.51	44.55	44.51
17	39.62	39.72	38.20	40.07	39.98	39.14	39.70	39.62	39.50	39.78	39.89	39.69
18	35.89	36.01	34.90	36.52	36.06	35.80	35.77	36.93	35.42	37.25	37.31	37.35
19	40.55	40.84	40.06	41.52	41.71	41.19	40.56	42.67	39.93	42.88	42.69	42.82
20	41.86	41.98	41.08	42.43	42.55	41.93	41.81	42.94	41.36	43.10	43.11	43.11
21	40.42	40.59	39.45	40.98	40.98	40.46	40.45	41.05	40.07	41.31	41.45	41.37
22	40.47	40.63	39.71	41.23	41.42	40.70	40.41	42.02	40.02	42.16	42.19	42.32
23	43.06	43.45	42.33	43.65	43.78	43.18	42.98	44.46	42.65	44.88	45.07	45.13
24	43.16	43.20	42.45	43.91	44.35	43.70	42.74	45.31	42.11	45.33	45.35	45.45
Ave.	41.10	41.18	40.26	41.79	41.92	41.28	41.21	42.20	40.60	42.50	42.76	42.74

subsampled G pixel values by Eq. (13). Then, the guided upsampling of the subsampled R pixel values is performed horizontally to obtain the horizontal tentative estimate \tilde{R}^H . In the horizontal guided upsampling, we use the horizontal sparse Laplacian filter $[-1 \ 0 \ 2 \ 0 \ -1]$. Subsequently, the residuals are calculated and horizontally interpolated. Finally, we estimate the horizontally interpolated \tilde{R}^H by adding the horizontal tentative estimate to the horizontally interpolated residual image.

In the step (ii), the color differences for horizontal and vertical directions are calculated as

$$\tilde{\delta}_{g,r}^H(i, j) = \begin{cases} \tilde{G}_{i,j}^V - R_{i,j}, & \text{at R pixel} \\ G_{i,j} - \tilde{R}_{i,j}^H, & \text{at G pixel,} \end{cases}$$

$$\tilde{\delta}_{g,r}^V(i, j) = \begin{cases} \tilde{G}_{i,j}^V - R_{i,j}, & \text{at R pixel} \\ G_{i,j} - \tilde{R}_{i,j}^V, & \text{at G pixel.} \end{cases} \quad (14)$$

The horizontal and vertical color differences of the B pixel are calculated in the same manner.

In the step (iii), the horizontal and vertical color differences are smoothed and combined as follows.

$$\begin{aligned} \tilde{\delta}_{g,r}(i, j) &= \{\omega_N * f_{NE} * \tilde{\delta}_{g,r}^V(i-3:i, j) \\ &\quad + \omega_S * f_{SW} * \tilde{\delta}_{g,r}^V(i:i+3, j) \\ &\quad + \omega_E * \tilde{\delta}_{g,r}^H(i, j-3:j) * f_{NE}^T \\ &\quad + \omega_W * \tilde{\delta}_{g,r}^H(i, j:j+3) * f_{SW}^T\} / \omega_T, \\ \omega_T &= \omega_N + \omega_S + \omega_E + \omega_W. \end{aligned} \quad (15)$$

TABLE V

S-CIELAB PERFORMANCE OF THE HIGH-RESOLUTION KODAK 24 IMAGES, WHERE BOLD TYPEFACE REPRESENTS THE BEST PERFORMANCE

No.	AP	SA	AHD	DLM	LPA	DFPD	GBTF	LDI-NAT	LSLC	Proposed		
										RI	MLRI	MLRI+wei.
1	0.703	0.732	0.782	0.661	0.648	0.698	0.690	0.610	0.721	0.608	0.608	0.604
2	0.992	1.047	1.364	0.991	0.982	1.001	1.032	0.719	1.066	0.483	0.428	0.413
3	0.545	0.556	0.608	0.516	0.514	0.571	0.531	0.508	0.569	0.523	0.497	0.502
4	0.652	0.702	0.821	0.657	0.619	0.680	0.690	0.557	0.718	0.554	0.511	0.509
5	0.642	0.722	0.757	0.627	0.599	0.633	0.727	0.523	0.759	0.533	0.527	0.524
6	0.585	0.606	0.640	0.542	0.540	0.589	0.581	0.501	0.583	0.516	0.530	0.526
7	0.576	0.600	0.644	0.543	0.542	0.609	0.563	0.538	0.583	0.551	0.532	0.541
8	1.329	1.338	1.414	1.212	1.237	1.286	1.256	1.165	1.310	1.144	1.141	1.148
9	0.862	0.866	0.916	0.799	0.804	0.918	0.797	0.840	0.820	0.876	0.837	0.859
10	0.817	0.827	0.877	0.759	0.768	0.869	0.771	0.790	0.784	0.833	0.782	0.799
11	0.606	0.631	0.679	0.564	0.558	0.602	0.611	0.512	0.631	0.523	0.521	0.519
12	0.404	0.411	0.436	0.376	0.378	0.420	0.385	0.379	0.404	0.433	0.380	0.383
13	1.006	1.048	1.195	0.944	1.032	0.950	1.114	0.786	1.087	0.763	0.766	0.755
14	0.639	0.676	0.715	0.600	0.569	0.633	0.639	0.509	0.716	0.518	0.505	0.503
15	0.404	0.434	0.490	0.389	0.368	0.399	0.428	0.338	0.454	0.358	0.334	0.332
16	0.637	0.653	0.703	0.602	0.601	0.641	0.639	0.565	0.646	0.582	0.582	0.579
17	0.839	0.835	0.910	0.782	0.801	0.880	0.821	0.779	0.824	0.786	0.764	0.769
18	1.397	1.396	1.548	1.286	1.378	1.414	1.393	1.194	1.420	1.206	1.154	1.147
19	0.834	0.832	0.891	0.739	0.728	0.780	0.833	0.643	0.860	0.660	0.674	0.665
20	0.625	0.635	0.694	0.592	0.588	0.627	0.646	0.542	0.659	0.552	0.551	0.546
21	0.846	0.868	0.958	0.814	0.825	0.841	0.896	0.744	0.877	0.740	0.747	0.739
22	0.789	0.819	0.926	0.743	0.726	0.785	0.825	0.658	0.820	0.656	0.671	0.658
23	0.556	0.549	0.625	0.532	0.523	0.554	0.571	0.477	0.581	0.451	0.457	0.455
24	0.589	0.617	0.683	0.567	0.550	0.582	0.647	0.492	0.640	0.501	0.516	0.508
Ave.	0.745	0.767	0.845	0.701	0.703	0.748	0.754	0.640	0.772	0.640	0.626	0.624

TABLE VI

CPSNR PERFORMANCE FOR THE ARRI 12 IMAGES, WHERE BOLD TYPEFACE REPRESENTS THE BEST PERFORMANCE

No.	AP	SA	AHD	DLM	LPA	DFPD	GBTF	LDI-NAT	LSLC	Proposed		
										RI	MLRI	MLRI+wei.
1	36.28	36.93	35.23	35.92	36.90	36.75	35.88	38.61	35.59	40.53	39.51	40.38
2	33.34	34.46	33.21	33.65	34.52	34.41	33.04	36.98	31.38	38.61	38.26	39.01
3	28.66	30.48	29.12	29.19	30.83	30.58	29.95	35.65	27.70	37.04	36.76	36.90
4	25.94	27.89	25.94	26.48	28.42	28.13	27.82	33.08	25.39	36.18	35.82	35.56
5	22.64	23.30	23.17	23.76	24.80	24.46	23.74	27.76	21.49	29.92	29.66	32.16
6	32.74	33.76	32.11	32.17	33.76	33.34	31.96	37.46	30.10	38.66	39.10	39.66
7	29.20	30.47	28.99	29.14	30.46	30.34	29.25	34.66	27.34	37.16	37.41	38.02
8	25.99	28.11	26.17	26.81	29.53	28.93	27.11	32.35	25.15	34.71	34.88	34.82
9	30.38	32.13	30.78	31.03	32.61	32.01	31.62	35.21	28.79	37.14	36.63	37.72
10	34.40	34.92	33.40	33.37	35.32	34.79	33.91	38.78	32.00	40.21	40.26	40.48
11	31.50	33.08	30.63	31.17	33.43	32.92	31.96	37.14	30.03	39.76	39.78	40.24
12	26.95	28.22	26.77	27.08	29.15	28.66	27.66	31.02	26.11	32.50	32.23	32.58
Ave.	29.83	31.15	29.63	29.98	31.64	31.28	30.33	34.89	28.42	36.87	36.69	37.29

TABLE VII

S-CIELAB PERFORMANCE OF THE ARRI 12 IMAGES, WHERE BOLD TYPEFACE REPRESENTS THE BEST PERFORMANCE

No.	AP	SA	AHD	DLM	LPA	DFPD	GBTF	LDI-NAT	LSLC	Proposed		
										RI	MLRI	MLRI+wei.
1	0.950	0.893	1.162	1.020	0.816	0.837	0.943	0.671	1.009	0.543	0.589	0.559
2	0.937	0.923	1.191	1.036	0.889	0.921	1.110	0.690	1.207	0.601	0.636	0.594
3	2.051	1.527	1.852	1.797	1.306	1.362	1.602	1.034	2.217	0.886	0.899	0.911
4	2.963	2.066	2.318	2.398	1.629	1.699	1.836	1.309	3.040	1.058	1.050	1.097
5	2.668	2.832	2.667	2.492	1.846	1.967	2.381	1.419	3.308	1.211	1.243	1.137
6	1.038	1.000	1.443	1.284	1.005	1.062	1.348	0.691	1.504	0.597	0.606	0.569
7	1.629	1.487	1.971	1.791	1.413	1.445	1.811	1.013	2.080	0.861	0.852	0.818
8	2.400	1.801	2.259	2.110	1.472	1.481	2.089	1.163	2.751	0.979	0.984	1.002
9	1.309	1.135	1.396	1.319	0.984	1.060	1.146	0.850	1.706	0.690	0.712	0.702
10	0.969	0.977	1.398	1.279	0.935	0.970	1.281	0.695	1.406	0.593	0.612	0.580
11	1.351	1.224	1.660	1.464	1.139	1.191	1.408	0.832	1.685	0.659	0.672	0.634
12	2.090	1.852	2.439	2.049	1.629	1.722	1.913	1.319	2.374	1.053	1.113	1.069
Ave.	1.696	1.476	1.813	1.670	1.255	1.310	1.572	0.974	2.024	0.811	0.831	0.806

In the GBTF [19] algorithm, a uniform averaging (boxcar) filter is used for smoothing the directional color differences in Eq. (15). In the proposed algorithm, we

introduce a Gaussian weighted averaging filter instead of the uniform averaging (boxcar) filter. We empirically use 1 for the standard deviation of the Gaussian

TABLE VIII

TOTAL AVERAGE CPSNR AND S-CIELAB PERFORMANCE OF ALL 54 IMAGES, WHERE BOLD TYPEFACE REPRESENTS THE BEST PERFORMANCE

No.	AP	SA	AHD	DLM	LPA	DFPD	GBTF	LDI-NAT	LSLC	Proposed		
										RI	MLRI	MLRI+wei.
PSNR	34.73	35.10	34.46	35.42	36.09	35.61	35.14	37.79	33.96	38.62	38.69	38.98
SCIELab	1.377	1.410	1.462	1.273	1.127	1.167	1.289	0.918	1.525	0.839	0.846	0.822

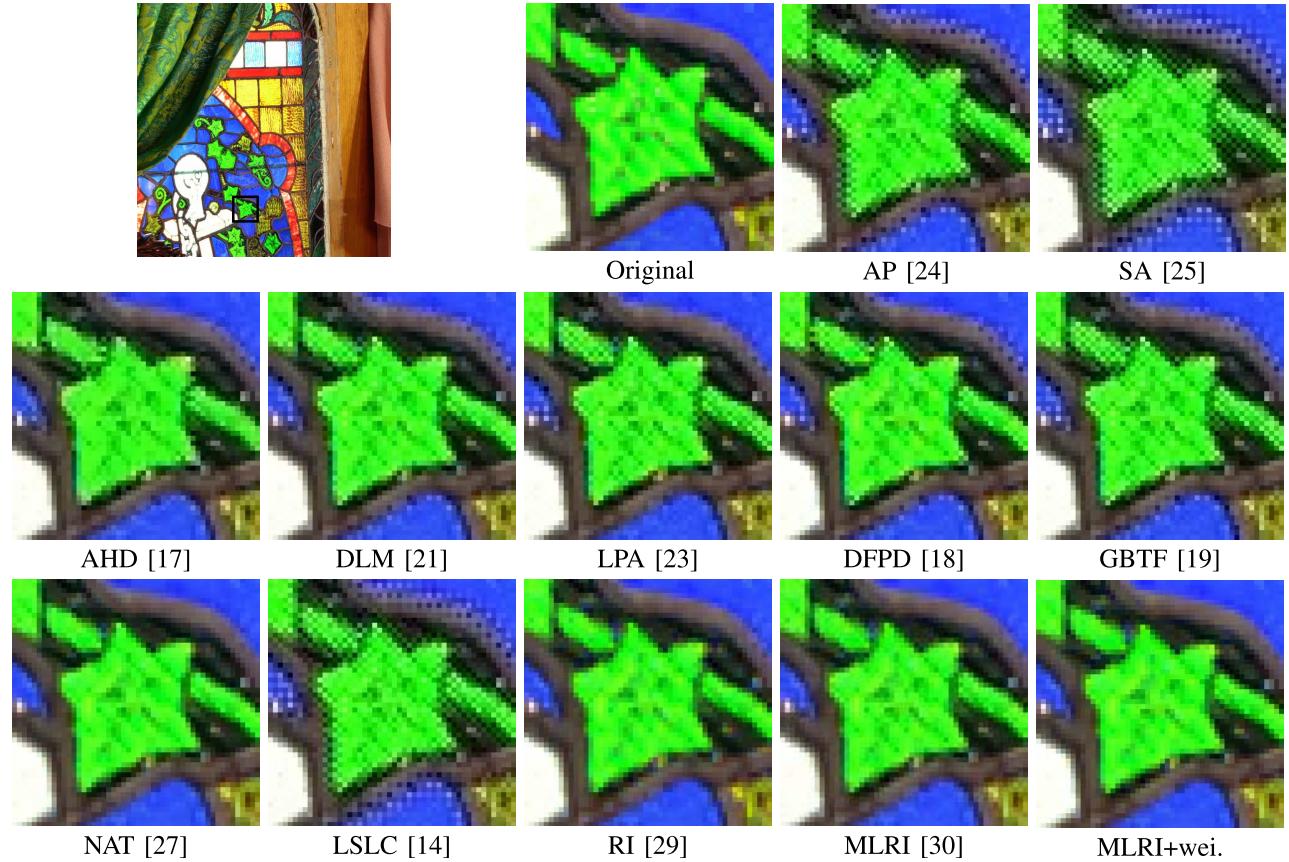


Fig. 8. Visual comparison for the star region in the IMAX dataset.

weight as

$$\begin{aligned} f_{NE} &= [0.01, 0.08, 0.35, 0.56], \\ f_{SW} &= [0.56, 0.35, 0.08, 0.01]. \end{aligned} \quad (16)$$

The weight for each direction ($\omega_N, \omega_S, \omega_E, \omega_W$) is calculated using color difference gradients in horizontal and vertical directions as

$$\begin{aligned} \omega_E &= 1/\left(\sum_{a=i-1}^{i+1} \sum_{b=j}^{j+2} D_{a,b}^H\right)^2, \quad \omega_W = 1/\left(\sum_{a=i-1}^{i+1} \sum_{b=j-2}^j D_{a,b}^H\right)^2, \\ \omega_N &= 1/\left(\sum_{a=i-2}^i \sum_{b=j-1}^{j+1} D_{a,b}^V\right)^2, \quad \omega_S = 1/\left(\sum_{a=i}^{i+2} \sum_{b=j-1}^{j+1} D_{a,b}^V\right)^2, \end{aligned} \quad (17)$$

where the directional gradients are calculated as

$$\begin{aligned} D_{i,j}^H &= \|\tilde{\delta}_{i,j-1}^H - \tilde{\delta}_{i,j+1}^H\|, \\ D_{i,j}^V &= \|\tilde{\delta}_{i-1,j}^V - \tilde{\delta}_{i+1,j}^V\|. \end{aligned} \quad (18)$$

Finally, in the step (iv), we obtain the interpolated G pixel value at the R or B pixel by adding the observed R or B pixel

value into the combined color difference as

$$\begin{aligned} \tilde{G}(i, j) &= R(i, j) + \tilde{\delta}_{g,r}(i, j), \\ \tilde{G}(i, j) &= B(i, j) + \tilde{\delta}_{g,b}(i, j). \end{aligned} \quad (19)$$

B. Red and Blue Interpolation

After the G image is interpolated, the GBTF algorithm interpolates the R and B pixel values by the standard color difference interpolation, as shown in Fig. 3 (a). We simply replace the color difference interpolation with the proposed MLRI as shown in Fig. 3 (b). We use bilinear interpolation for the interpolation of residuals.

V. EXPERIMENTAL RESULTS

A. Comparison of the Color Difference Interpolation and the Residual Interpolation

We first demonstrate the effectiveness of our proposed tentative estimate generation. Fig. 6 shows examples of (a) RGB image, (b) G image, (c) R image, (d) Laplacian map of the R image, (e) color difference image, (f) Laplacian map of the color difference image, (g) tentative estimate of the R image, (h) residual image, and (i) Laplacian map of the

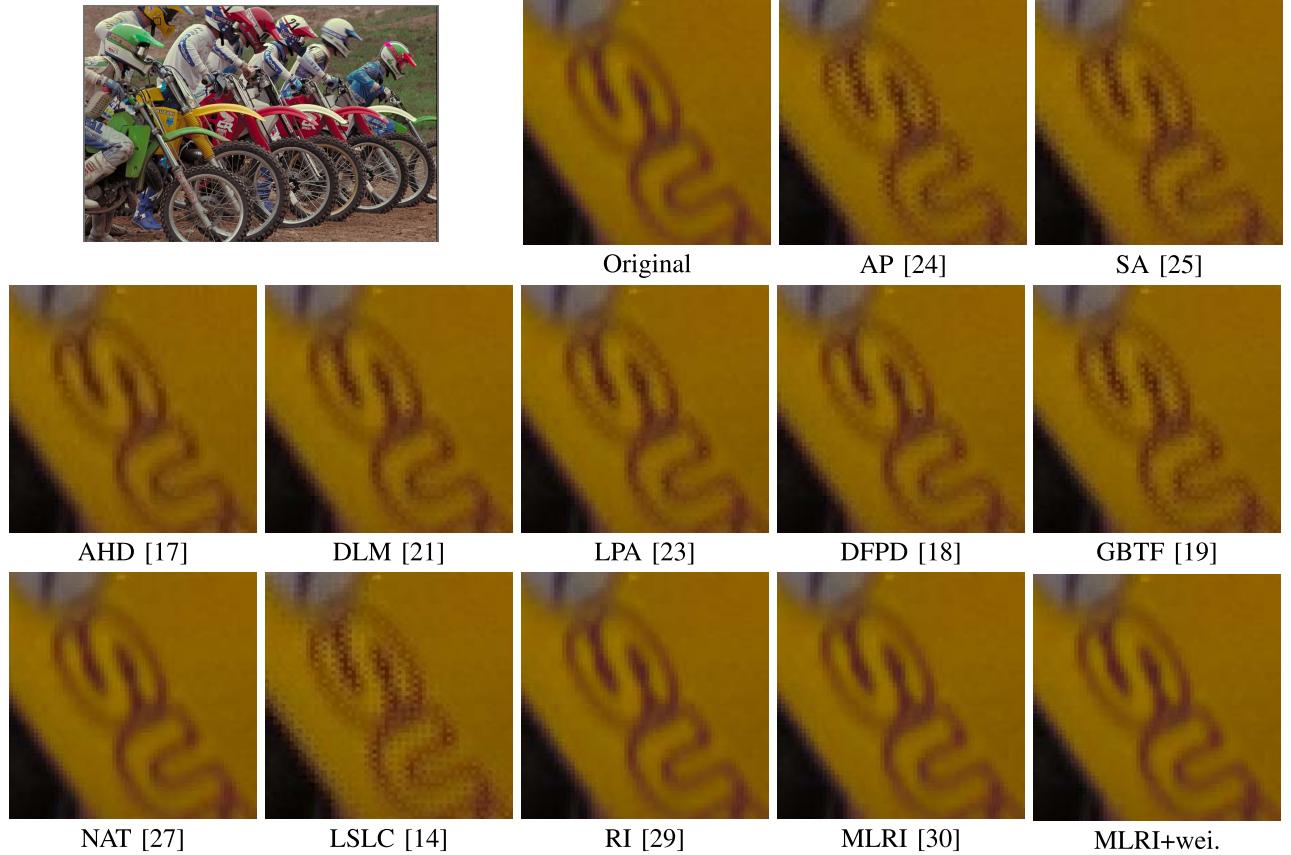


Fig. 9. Visual comparison of characters in the high-resolution Kodak dataset.

residual image. This image is from the IMAX dataset [27]. Here, the tentative estimate of the R image was generated by the guided upsampling using the ground-truth G image to remove effects of interpolation errors for purely showing the nature of the residual calculation. The average intensities of the color difference image and the residual image are normalized for visualization in Fig. 6 (e) and (h). Fig. 6 (j), (k), and (l) shows the sliced plots of pixel values on the white lines shown in Fig. 6 (a). Fig. 6 (j) shows the sliced plots of the original R, the original G, and the tentative estimate images. Fig. 6 (k) shows the sliced plot of the color difference image and Fig. 6 (l) shows that of the residual image. From these results, we can see that our proposed tentative estimation can effectively reduce the Laplacian energy of the residuals and generate smoother residual image compared with the color difference image.

We next evaluate the interpolation performance in the residual domain. In the evaluation, we subsampled the original R and the original B images to have the Bayer sampled pattern and interpolated the subsampled images using bilinear interpolation in the three different domains (channel-independent, color difference, and residual). The ground-truth G image was used for the calculation of the color difference and the residual to purely evaluate domain differences. Fig. 7 (a) and (c) show the PSNR performances of the channel-independent interpolation, the color difference interpolation, and the proposed MLRI for the R and the B bands of the IMAX 18 images [27].

Fig. 7 (b) and (d) show the corresponding approximate Laplacian energies of the R and the B images, the color difference images, and the residual images. In these results, the proposed MLRI outperforms the other two interpolation methods by reducing the approximate Laplacian energy of the residuals. For the B band of the image number 16 and 17, the approximate Laplacian energy of the color difference image is larger than that of the B image itself. Consequently, the PSNR performance of the color difference interpolation is lower than that of the channel-independent interpolation for the B band of the image number 16 and 17. These results validate the usefulness of the Laplacian energy minimization for the interpolation.

B. Comparison With State-of-the-Art Algorithms

The proposed algorithm¹ was evaluated using three full color image datasets, the IMAX [27] dataset, the Kodak dataset [32]–[34], and the beyond Kodak dataset [35]. The IMAX dataset [27] consists of 18 images. The image size is 500×500 . The IMAX images are cropped from original 2310×1814 high-resolution images. The Kodak dataset consists of 24 images. There are several types of Kodak dataset with different resolutions. The original one consists of 3072×2048 high-resolution images [33], [34], while the low-resolution version consists of downsampled 768×512 images [32]. As noted in [35], the low-resolution

¹Source code available: <http://www.ok.ctrl.titech.ac.jp/res/DM/RI.html>

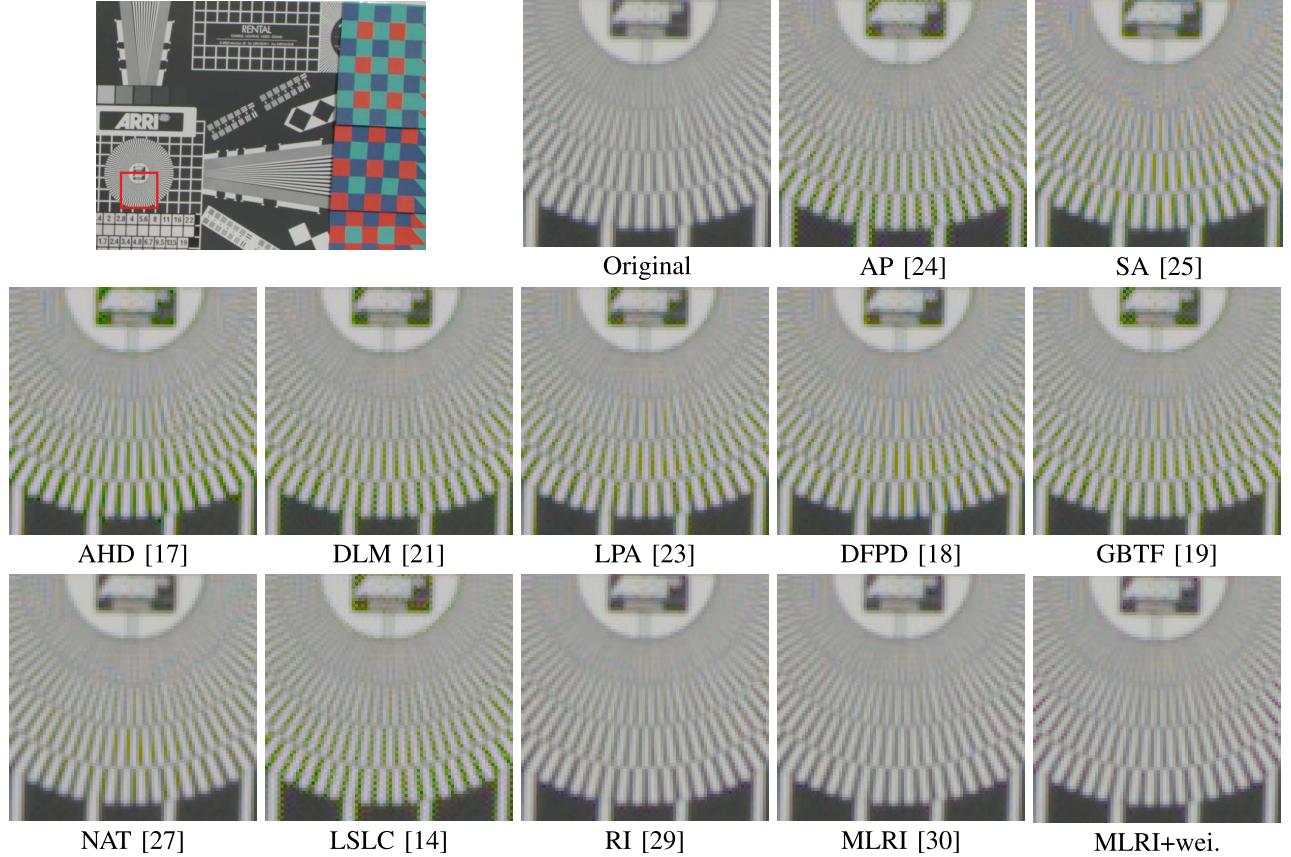


Fig. 10. Visual comparison for the chart in the ARRI dataset.

version is widely used for the evaluation in existing works because of the limitation of computational power in spite of undesirable downsampling effects. In this paper, we used the high-resolution version because the computational power of current PCs has been much improved and it is not a big problem. The beyond Kodak dataset [35], which we call the ARRI dataset in the following, is a new high-quality image dataset. The ARRI dataset is expected as an alternative to the Kodak dataset. The ARRI dataset contains one 2880×1620 high-resolution full color image captured using a color wheel. We divided the full color image into 12 sub-images, where the image size is 720×540 . Table I summarizes dataset statistics evaluated in [27]. The high-performance demosaicking algorithm for various datasets is one of the most desirable properties.

We compared the proposed algorithm with state-of-the-art algorithms; alternating projections (AP) [24], successive approximation (SA) [25], adaptive homogeneity-directed (AHD) [17], directional linear minimum mean square-error estimation (DLMSE) [21], local polynomial approximation (LPA) [23], directional filtering and a posteriori decision (DFPD) [18], gradient based threshold free (GBTF) [19], local directional interpolation and nonlocal adaptive thresholding (NAT) [27], and least-squares lumachroma (LSLC) [14]. We implemented the GBTF algorithm because it is not publicly available. The other source codes are downloaded from the author's websites. The RI and

MLRI are our proposed RI algorithms in [29] and [30]. The MLRI+wei. is the algorithm described in this paper, which introduces the weighted averaging of the GF.

We evaluated CPSNR and S-CIELAB [39], [40] values, which are used in the representative survey paper [4]. Table II shows the CPSNR performance and Table III shows the S-CIELAB performance for the IMAX 18 images. Table IV shows the CPSNR performance and Table V shows the S-CIELAB performance for the high-resolution Kodak 24 images. Table VI shows the CPSNR performance and Table VII shows the S-CIELAB performance for the ARRI 12 images. Our proposed algorithm generally outperforms existing state-of-the-art algorithms on average for the IMAX, the high-resolution Kodak, and the ARRI datasets. It is remarkable that several algorithms only work well for one dataset, but do not for another dataset. For example, the GBTF and the LPA algorithms only work well for the Kodak dataset. Table VIII shows the average CPSNR and S-CIELAB of all 54 images. The proposed algorithm outperforms all state-of-the-art algorithms in terms of the total average CPSNR and S-CIELAB, which validates the effectiveness of our proposed algorithm for various image datasets.

Fig. 8 shows visual comparison of the star region in the IMAX dataset. Fig. 9 shows visual comparisons of the characters in the high-resolution Kodak dataset. In these results, our proposed algorithm effectively can reduce zipper artifacts. The NAT algorithm also works well for these scenes. Fig. 10

shows visual comparison of the chart in the ARRI dataset. For the chart scene, our proposed algorithm can sharply interpolate the image without color artifacts, while the other algorithms, including the NAT algorithm, generate severe color artifacts.

The computational time of our proposed algorithm is reasonably fast and takes about 1.61 seconds for a 500×500 image in MATLAB implementation on Windows desktop PC with an Intel Xeon E5-1603 v3 2.80 GHz processor and 16.0 GB RAM.

VI. CONCLUSION

In this paper, we proposed the RI as an alternative to widely used color difference interpolation for color image demosaicking. We experimentally showed that the interpolation accuracy is improved by reducing the Laplacian energy of the image to be interpolated. Based on this observation, we proposed the MLRI, which performs the interpolation in the residual domain with the minimized Laplacian energy, where the residuals are differences between the observed and the tentatively estimated pixel values. We estimate the tentative pixel values by minimizing the Laplacian energy of the residuals by guided upsampling. We also proposed a novel demosaicking algorithm by incorporating the proposed MLRI into the GBTF algorithm, which is one of state-of-the-art Bayer demosaicking algorithms. Experimental results demonstrate that our proposed demosaicking algorithm using the MLRI can offer state-of-the-art results with reduced color artifacts for the IMAX, the Kodak, and the ARRI datasets. Our future works include the consideration of noise effects in our proposed algorithm and the extension of our proposed algorithm for another type of CFAs [41], [42].

REFERENCES

- [1] R. Lukac, *Single-Sensor Imaging: Methods and Applications for Digital Cameras*. Boca Raton, FL, USA: CRC Press, 2008.
- [2] B. E. Bayer, "Color imaging array," U.S. Patent 3971065, Jul. 20, 1976.
- [3] B. K. Gunturk, J. Glotzbach, Y. Altunbasak, R. W. Schafer, and R. M. Mersereau, "Demosaicking: Color filter array interpolation," *IEEE Signal Process. Mag.*, vol. 22, no. 1, pp. 44–54, Jan. 2005.
- [4] X. Li, B. Gunturk, and L. Zhang, "Image demosaicing: A systematic survey," *Proc. SPIE*, vol. 6822, pp. 68221J-1–68221J-15, Jan. 2008.
- [5] D. Menon and G. Calvagno, "Color image demosaicking: An overview," *Signal Process., Image Commun.*, vol. 26, nos. 8–9, pp. 518–533, Oct. 2011.
- [6] R. Kimmel, "Demosaicing: Image reconstruction from color CCD samples," *IEEE Trans. Image Process.*, vol. 8, no. 9, pp. 1221–1228, Sep. 1999.
- [7] R. Lukac and K. N. Plataniotis, "Normalized color-ratio modeling for CFA interpolation," *IEEE Trans. Consum. Electron.*, vol. 50, no. 2, pp. 737–745, May 2004.
- [8] D. R. Cok, "Signal processing method and apparatus for producing interpolated chrominance values in a sampled color image signal," U.S. Patent 4642678, Feb. 10, 1987.
- [9] R. H. Hibbard, "Apparatus and method for adaptively interpolating a full color image utilizing luminance gradients," U.S. Patent 5382976, Jan. 17, 1995.
- [10] J. F. Hamilton, Jr., and J. E. Adams, Jr., "Adaptive color plan interpolation in single sensor color electronic camera," U.S. Patent 5629734, May 13, 1997.
- [11] D. Alleysson, S. Stüsstrunk, and J. Héault, "Linear demosaicing inspired by the human visual system," *IEEE Trans. Image Process.*, vol. 14, no. 4, pp. 439–449, Apr. 2005.
- [12] E. Dubois, "Frequency-domain methods for demosaicking of Bayer-sampled color images," *IEEE Signal Process. Lett.*, vol. 12, no. 12, pp. 847–850, Dec. 2005.
- [13] N.-X. Lian, L. Chang, Y.-P. Tan, and V. Zagorodnov, "Adaptive filtering for color filter array demosaicking," *IEEE Trans. Image Process.*, vol. 16, no. 10, pp. 2515–2525, Oct. 2007.
- [14] B. Leung, G. Jeon, and E. Dubois, "Least-squares luma-chroma demultiplexing algorithm for Bayer demosaicking," *IEEE Trans. Image Process.*, vol. 20, no. 7, pp. 1885–1894, Jul. 2011.
- [15] J. Mairal, F. Bach, J. Ponce, G. Sapiro, and A. Zisserman, "Non-local sparse models for image restoration," in *Proc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Sep./Oct. 2009, pp. 2272–2279.
- [16] A. A. Moghadam, M. Aghagolzadeh, M. Kumar, and H. Radha, "Compressive framework for demosaicing of natural images," *IEEE Trans. Image Process.*, vol. 22, no. 6, pp. 2356–2371, Jun. 2013.
- [17] K. Hirakawa and T. W. Parks, "Adaptive homogeneity-directed demosaicing algorithm," *IEEE Trans. Image Process.*, vol. 14, no. 3, pp. 360–369, Mar. 2005.
- [18] D. Menon, S. Andriani, and G. Calvagno, "Demosaicing with directional filtering and *a posteriori* decision," *IEEE Trans. Image Process.*, vol. 16, no. 1, pp. 132–141, Jan. 2007.
- [19] I. Pekkucuksen and Y. Altunbasak, "Gradient based threshold free color filter array interpolation," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2010, pp. 137–140.
- [20] I. Pekkucuksen and Y. Altunbasak, "Multiscale gradients-based color filter array interpolation," *IEEE Trans. Image Process.*, vol. 22, no. 1, pp. 157–165, Jan. 2013.
- [21] L. Zhang and X. Wu, "Color demosaicing via directional linear minimum mean square-error estimation," *IEEE Trans. Image Process.*, vol. 14, no. 12, pp. 2167–2178, Dec. 2005.
- [22] K.-H. Chung and Y.-H. Chan, "Color demosaicing using variance of color differences," *IEEE Trans. Image Process.*, vol. 15, no. 10, pp. 2944–2955, Oct. 2006.
- [23] D. Paliy, V. Katkovnik, R. Bilcu, S. Alenius, and K. Egiazarian, "Spatially adaptive color filter array interpolation for noiseless and noisy data," *Int. J. Imag. Syst. Technol.*, vol. 17, no. 3, pp. 105–122, 2007.
- [24] B. K. Gunturk, Y. Altunbasak, and R. M. Mersereau, "Color plane interpolation using alternating projections," *IEEE Trans. Image Process.*, vol. 11, no. 9, pp. 997–1013, Sep. 2002.
- [25] X. Li, "Demosaicing by successive approximation," *IEEE Trans. Image Process.*, vol. 14, no. 3, pp. 370–379, Mar. 2005.
- [26] A. Buades, B. Coll, J.-M. Morel, and C. Sbert, "Self-similarity driven color demosaicing," *IEEE Trans. Image Process.*, vol. 18, no. 6, pp. 1192–1202, Jun. 2009.
- [27] L. Zhang, X. Wu, A. Buades, and X. Li, "Color demosaicing by local directional interpolation and nonlocal adaptive thresholding," *J. Electron. Imag.*, vol. 20, no. 2, pp. 023016-1–023016-16, Jun. 2011.
- [28] J. Duran and A. Buades, "Self-similarity and spectral correlation adaptive algorithm for color demosaicing," *IEEE Trans. Image Process.*, vol. 23, no. 9, pp. 4031–4040, Sep. 2014.
- [29] D. Kiku, Y. Monno, M. Tanaka, and M. Okutomi, "Residual interpolation for color image demosaicking," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2013, pp. 2304–2308.
- [30] D. Kiku, Y. Monno, M. Tanaka, and M. Okutomi, "Minimized-Laplacian residual interpolation for color image demosaicking," *Proc. SPIE*, vol. 9023, pp. 90230L-1–90230L-8, Mar. 2014.
- [31] K. He, J. Sun, and X. Tang, "Guided image filtering," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 6, pp. 1397–1409, Jun. 2013.
- [32] True Color Kodak Images. [Online]. Available: <http://r0k.us/graphics/kodak/>
- [33] Bradley J. Lucier's Page. [Online]. Available: <http://www.math.purdue.edu/~lucier/>
- [34] Kodak Image Dataset (High-Resolution Version). [Online]. Available: http://www.math.purdue.edu/~lucier/PHOTO_CD/BMP_IMAGES/
- [35] S. Andriani, H. Brendel, T. Seybold, and J. Goldstone, "Beyond the Kodak image set: A new reference set of color image sequences," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2013, U.S. Patent 2289–2293.
- [36] C. Tomasi and R. Manduchi, "Bilateral filtering for gray and color images," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Jan. 1998, pp. 839–846.
- [37] Y. Monno, M. Tanaka, and M. Okutomi, "Multispectral demosaicking using adaptive kernel upsampling," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2011, pp. 3157–3160.
- [38] Y. Monno, M. Tanaka, and M. Okutomi, "Multispectral demosaicking using guided filter," *Proc. SPIE*, vol. 8299, pp. 82990O-1–82990O-7, Jan. 2012.

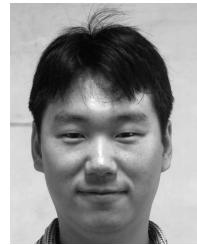
- [39] *S-CIELAB Page*. [Online]. Available: <http://white.stanford.edu/~brian/scielab/>
- [40] X. Zhang and B. A. Wandell, "A spatial extension of CIELAB for digital color-image reproduction," *J. Soc. Inf. Display*, vol. 5, no. 1, pp. 61–63, Mar. 1997.
- [41] K. Hirakawa and P. J. Wolfe, "Spatio-spectral color filter array design for optimal image recovery," *IEEE Trans. Image Process.*, vol. 17, no. 10, pp. 1876–1890, Oct. 2008.
- [42] L. Condat, "A new color filter array with optimal properties for noiseless and noisy color image acquisition," *IEEE Trans. Image Process.*, vol. 20, no. 8, pp. 2200–2210, Aug. 2011.



Daisuke Kiku received the B.E. and M.E. degrees from the Tokyo Institute of Technology, Tokyo, Japan, in 2013, and 2015, respectively. He is currently a Senior Developer with Olympus Corporation, where he is engaged in the development of gastrointestinal endoscope at the Endoscope Development Department. His work focuses on multibending endoscope.



Yusuke Monno (M'14) received the B.E., M.E., and Ph.D. degrees from the Tokyo Institute of Technology, Tokyo, Japan, in 2010, 2011, and 2014, respectively. He is currently a Post-Doctoral Researcher at the Tokyo Institute of Technology. From 2013 to 2014, he was with the Image and Visual Representation Group, École Polytechnique Fédérale de Lausanne, Switzerland, as a Research Internship Student. His research interests are in both theoretical and practical aspects of computer vision and image processing.



University, CA, USA.

Masayuki Tanaka received the bachelor's and master's degrees in control engineering and Ph.D. degree from the Tokyo Institute of Technology, Tokyo, Japan, in 1998, 2000, and 2003, respectively. He joined Agilent Technology in 2003. He was a Research Scientist with the Tokyo Institute of Technology from 2004 to 2008. Since 2008, he has been an Associated Professor with the Graduate School of Science and Engineering, Tokyo Institute of Technology. He was a Visiting Scholar with the Department of Psychology, Stanford



Masatoshi Okutomi received the B.Eng. degree from the Department of Mathematical Engineering and Information Physics, The University of Tokyo, Tokyo, Japan, in 1981, the M.Eng. degree from the Department of Control Engineering, Tokyo Institute of Technology, Tokyo, in 1983, and the D.Eng. degree from the Tokyo Institute of Technology, in 1993, for his research on stereo vision.. He joined the Canon Research Center, Canon Inc., Tokyo, in 1983. From 1987 to 1990, he was a Visiting Research Scientist with the School of Computer Science, Carnegie Mellon University, Pittsburgh, PA, USA. Since 1994, he has been with the Tokyo Institute of Technology, where he is currently a Professor with the Department of Mechanical and Control Engineering, Graduate School of Science and Engineering.