ADAPTIVE RESIDUAL INTERPOLATION FOR COLOR IMAGE DEMOSAICKING

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

Tokyo Institute of Technology, Tokyo, Japan

ABSTRACT

Color image demosaicking is an essential image processing operation for acquiring high-quality color images. Recently, demosaicking algorithms using residual interpolation (RI), which performs the interpolation in a residual domain, have been proposed. An iterative framework has also been introduced into the RI and shown state-of-the-art performance. In this paper, we propose a novel demosaicking algorithm using adaptive residual interpolation (ARI), which adaptively selects a suitable iteration number and combines two different types of RI algorithms at each pixel. Experimental results demonstrate that our demosaicking algorithm can achieve a clear improvement in comparison with existing algorithms.

***Index Terms*—** Bayer color filter array (CFA), demosaicking, residual interpolation, iterative framework.

1. INTRODUCTION

A single image sensor with a color filter array (CFA) is widely used in current color digital cameras, where only one pixel value among RGB values is recorded at each pixel [1]. The other two missing pixel values must be generated by an interpolation process, which is typically called a demosaicking process [2–4]. The demosaicking process plays a crucial role in acquiring high-quality color images.

The most popular and widely used CFA is the Bayer CFA [5]. Demosaicking algorithms for the Bayer CFA have extensively been studied [6–32]. The representative techniques include color ratio interpolation [6, 7], color difference interpolation [8–10], a frequency domain approach [11–14], and a non-local approach [15–17]. We refer to survey papers [2–4] for comprehensive reviews because it is too diverse to explain all existing algorithms here.

Recently, demosaicking algorithms using residual interpolation (RI) have been proposed [30–32]. The RI performs the interpolation in a residual domain, where the residual is defined as the difference between a tentatively estimated pixel value and an observed pixel value. The original RI [30] generates the tentative estimate by minimizing the residual itself using the guided filtering (GF) [33]. Its extended version called minimized-Laplacian RI (MLRI) [31] minimizes a Laplacian energy of the residual, instead of the residual itself. The iterative RI (IRI) [32], which iteratively updates the tentative estimate, has also been proposed and shown state-of-the-art performance.

In this paper, we propose adaptive residual interpolation (ARI) for color image demosaicking with the Bayer CFA. Main contributions of the ARI are; (i) the ARI adaptively selects a suitable iteration number for each pixel, instead of using a common iteration number decided for whole image pixels as conducted in [32], and (ii) the ARI adaptively combines the RI and the MLRI at each pixel to take full advantage of both RI algorithms. We extensively compare our proposed demosaicking algorithm using the ARI with existing algorithms and demonstrate that our proposed algorithm can achieve a clear improvement, about 0.5dB in CPSNR, for standard IMAX and Kodak 30 images.

2. PROPOSED DEMOSAICKING ALGORITHM

The proposed demosaicking algorithm firstly interpolates the missing G pixel values as most of existing algorithms do. In the following, we describe the G interpolation process by our proposed ARI. For the R and B interpolation processes, we simply follow the same way as the MLRI [31].

2.1. Overview of our proposed G interpolation.

Fig. 1 illustrates the overall flow of the G interpolation by our proposed ARI. We here only explains the G interpolation at the R pixels. The G interpolation at the B pixels is similarly performed.

The proposed ARI consists of three steps. (i) The G interpolation at the R lines is performed in both horizontal and vertical directions by the RI and the MLRI with the iterative framework. As the result of each directional interpolation, a set of directionally interpolated G images, where one image corresponds to one iteration, is generated. (ii) For each directional interpolation results, a suitable iteration number is selected from the set of directionally interpolated G images adaptively at each pixel. (iii) The results of the directional RI and MLRI are adaptively combined by a weighted averaging at each R pixel to obtain the final G interpolation result. We describe the details of each step in the following.

2.2. Step (i): Iterative directional interpolation.

In the step (i), the directional interpolation is performed by the RI and the MLRI with the iterative framework. Fig. 2

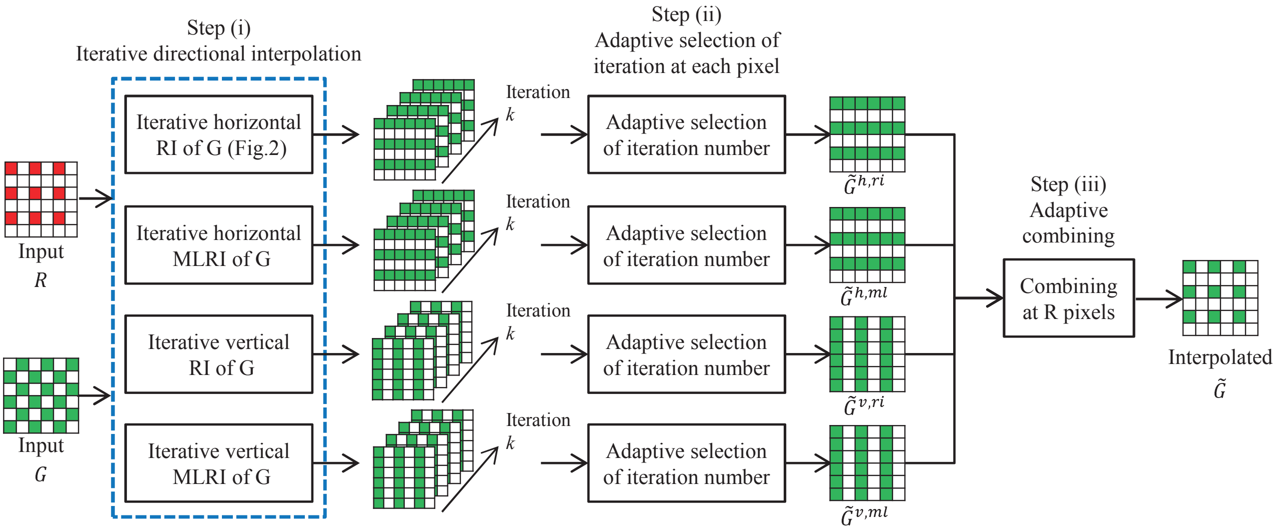

Fig. 1. Overall flow of the G interpolation by our proposed ARI.

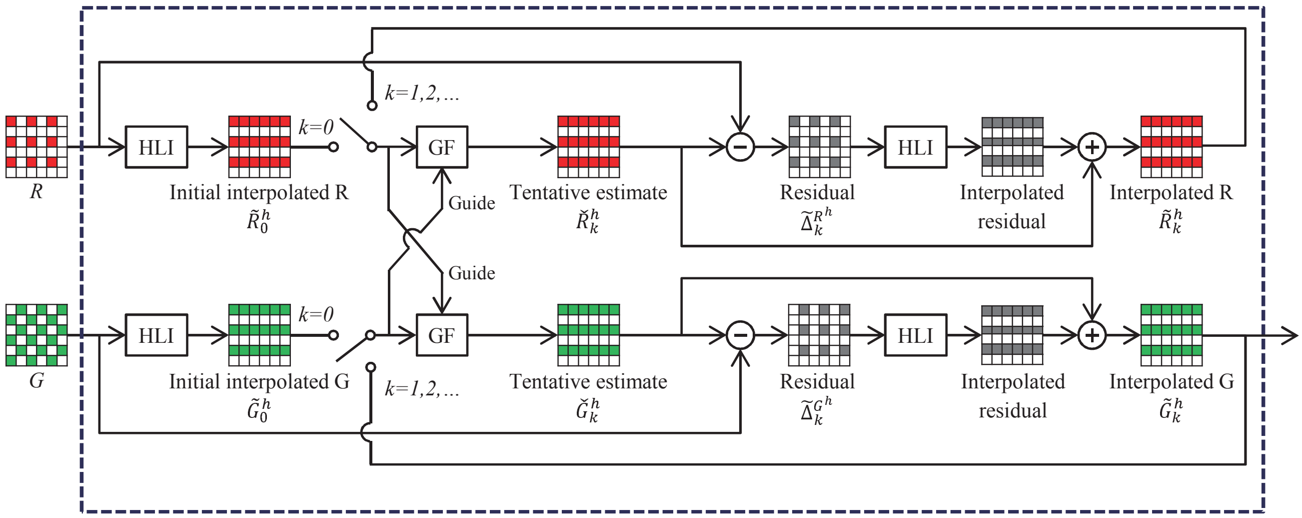

Fig. 2. Flow of the iterative horizontal G interpolation at the R lines by the RI (top block of the step (i) in Fig. 1).

illustrates the flow of the G interpolation at the R lines in the horizontal direction by the RI (top block of the step (i) in Fig. 1). The G interpolation in the vertical direction is similarly performed.

First, horizontal linear interpolation (HLI) of the R and the G at the R lines is performed to obtain initial interpolation results as

$$\begin{aligned}\tilde{R}^h_{(i,j),0} &= (R_{(i-1,j)} + R_{(i+1,j)})/2, \ \text{ at G pixel},\\ \tilde{G}^h_{(i,j),0} &= (G_{(i-1,j)} + G_{(i+1,j)})/2, \ \text{ at R pixel},\end{aligned} \tag{1}$$

where $\tilde{R}$ and $\tilde{G}$ represent interpolated pixel values, the subscript $(i, j), 0$ represents the initial interpolation result at a pixel (i, j), and the superscript h represents the horizontal direction. Tentative estimates of the R and the G at the k-th iteration are generated by linear transformations of the previous interpolation results (if $k = 0$, initial interpolation results are used) as

$$\begin{aligned}\check{R}^h_{(i,j),k} &= a^r_{(i,j),k}\tilde{G}^h_{(i,j),k-1} + b^r_{(i,j),k} \ ,\\ \check{G}^h_{(i,j),k} &= a^g_{(i,j),k}\tilde{R}^h_{(i,j),k-1} + b^g_{(i,j),k} \ ,\end{aligned} \tag{2}$$

where $\check{R}$ and $\check{G}$ represent tentatively estimated pixel values, and (a^r, b^r) and (a^g, b^g) are linear coefficients. The linear coefficients are calculated by using the GF [33], where $\tilde{G}$ is used as the guide for generating $\check{R}$ and vice versa. Specifically, the GF calculates the linear coefficients by minimizing the following error between the previous interpolation results and the tentative estimates at each local window.

$$\begin{aligned}d^{R^h}_{(i,j),k} &= \check{R}^h_{(i,j),k} - \tilde{R}^h_{(i,j),k-1} \ ,\\ d^{G^h}_{(i,j),k} &= \check{G}^h_{(i,j),k} - \tilde{G}^h_{(i,j),k-1} \ .\end{aligned} \tag{3}$$

Then, residuals are calculated as

$$\begin{aligned}\tilde{\Delta}^{R^h}_{(i,j),k} &= R_{(i,j)} - \check{R}^h_{(i,j),k}\ , \ \text{ at R pixel,}\\ \tilde{\Delta}^{G^h}_{(i,j),k} &= G_{(i,j)} - \check{G}^h_{(i,j),k}\ , \ \text{ at G pixel.}\end{aligned} \tag{4}$$

Next, the residuals are interpolated by the HLI as

$$\begin{aligned}\tilde{\Delta}^{R^h}_{(i,j),k} &= (\tilde{\Delta}^{R^h}_{(i-1,j),k} + \tilde{\Delta}^{R^h}_{(i+1,j),k})/2, \ \text{ at G pixel,}\\ \tilde{\Delta}^{G^h}_{(i,j),k} &= (\tilde{\Delta}^{G^h}_{(i-1,j),k} + \tilde{\Delta}^{G^h}_{(i+1,j),k})/2, \ \text{ at R pixel.}\end{aligned} \tag{5}$$

Finally, the tentative estimates are added to obtain k-th horizontally interpolated results as

$$\begin{aligned}\tilde{R}^h_{(i,j),k} &= \tilde{\Delta}^{R^h}_{(i,j),k} + \check{R}^h_{(i,j),k}\ ,\\ \tilde{G}^h_{(i,j),k} &= \tilde{\Delta}^{G^h}_{(i,j),k} + \check{G}^h_{(i,j),k}\ .\end{aligned} \tag{6}$$

In the iterative framework [32], the tentative estimates are iteratively updated by using previous interpolation results in Eq. (2). In [32], the iteration is globally stopped. In other words, a common iteration number is used for whole image pixels. In contrast, the proposed ARI generates a set of directionally interpolated images, where one image corresponds to one iteration, and adaptively selects a suitable iteration number for each pixel in the step (ii).

The directional interpolation by the MLRI is also performed by the same framework. The only difference is the way of tentative estimate generation. The MLRI calculate the linear coefficients in Eq.(2) by minimizing a Laplacian energy of the residual by using a modified GF. We refer to the original paper [31] for technical details.

2.3. Step (ii): Adaptive selection of iteration number.

In the step (ii), for each directional interpolation, the suitable iteration number is adaptively selected at each pixel based on a criteria [32]. We here only explain the criteria for the horizontal direction. The criteria for the vertical direction is similarly calculated. We define the criteria in a pixel by pixel manner, instead of the global manner used in [32], based on the absolute values and gradients of the GF errors in Eq. (3) as

$$c^h_{(i,j),k} = d^h_{(i,j),k} \cdot \delta d^h_{(i,j),k}, \tag{7}$$

where

$$d^h_{(i,j),k} = \left(|d^{R^h}_{(i,j),k}| + |d^{G^h}_{(i,j),k}|\right)^2, \tag{8}$$

$$\delta d^h_{(i,j),k} = |d^{R^h}_{(i-1,j),k} - d^{R^h}_{(i+1,j),k}| + |d^{G^h}_{(i-1,j),k} - d^{G^h}_{(i+1,j),k}|. \tag{9}$$

We select the suitable iteration number k_{opt}, which minimizes the criteria, adaptively at each pixel as

$$k_{opt} = \arg\min_k c^h_{(i,j),k}\ , \tag{10}$$

where we empirically used $k = 11$ for the maximum iteration number. Hereafter, we remove the subscript k, which means the suitable iteration number k_{opt} is already selected.

2.4. Step (iii): Adaptive combining of the RI algorithms.

In the step (iii), directional interpolation results of the RI and the MLRI are combined by the weighted averaging as

$$\tilde{G}_{(i,j)} = \frac{w^{h,ri}_{(i,j)}\tilde{G}^{h,ri}_{(i,j)} + w^{h,ml}_{(i,j)}\tilde{G}^{h,ml}_{(i,j)} + w^{v,ri}_{(i,j)}\tilde{G}^{v,ri}_{(i,j)} + w^{v,ml}_{(i,j)}\tilde{G}^{v,ml}_{(i,j)}}{w^{h,ri}_{(i,j)} + w^{h,ml}_{(i,j)} + w^{v,ri}_{(i,j)} + w^{v,ml}_{(i,j)}}, \tag{11}$$

where h and v represent the horizontal and the vertical directions, ri and ml represent the results of the RI and the MLRI respectively. The each weight is calculated based on the criteria as

$$\begin{aligned} w^{h,ri}_{(i,j)} &= 1/c^{h,ri}_{(i,j)}, & w^{h,ml}_{(i,j)} &= 1/c^{h,ml}_{(i,j)},\\ w^{v,ri}_{(i,j)} &= 1/c^{v,ri}_{(i,j)}, & w^{v,ml}_{(i,j)} &= 1/c^{v,ml}_{(i,j)},\end{aligned} \tag{12}$$

where a small criteria value contributes a large weight.

3. EXPERIMENTAL RESULTS

The proposed algorithm[1] is evaluated with two standard color image datasets, the IMAX dataset and the Kodak dataset used in [3]. The IMAX dataset consists of 18 images and the image size is 500×500. The Kodak dataset consists of 12 images and the image size is 768×512. We extensively compare the proposed demosaicking algorithm with 16 existing algorithms[2], including the three existing RI algorithms, i.e., RI [30], MLRI [31], and IRI [32]. Table 1 summarizes the average PSNR and CPSNR performances. The average PSNR and CPSNR of the proposed algorithm for the IMAX dataset outperforms all the existing algorithms. The average PSNR and CPSNR of the proposed algorithm for the Kodak dataset is lower than some of existing algorithms. However, it is remarkable that several algorithms only work well for one dataset, but do not for another dataset. The proposed algorithm outperforms all the state-of-the-art algorithms in terms of the total average PSNR and CPSNR of both datasets. Fig. 3 shows the visual comparison of the demosaicking results for the flower region in the IMAX dataset. From the visual comparison, we can find that the proposed algorithm can sharply interpolate the image without severe color and zipper artifacts.

4. CONCLUSION

In this paper, we have proposed the ARI for color image demosaicking. The ARI adaptively selects a suitable iteration number and combines two different types of RI algorithms, i.e., the RI and the MLRI, at each pixel. Experimental results demonstrate that our demosaicking algorithm with the ARI can achieve a clear improvement in extensive comparison with existing algorithms. Future work includes the extension of the ARI for multispectral demosaicking [34, 35].

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

[2] Only the GBTF algorithm is our implementation because the source code is not publicly available.

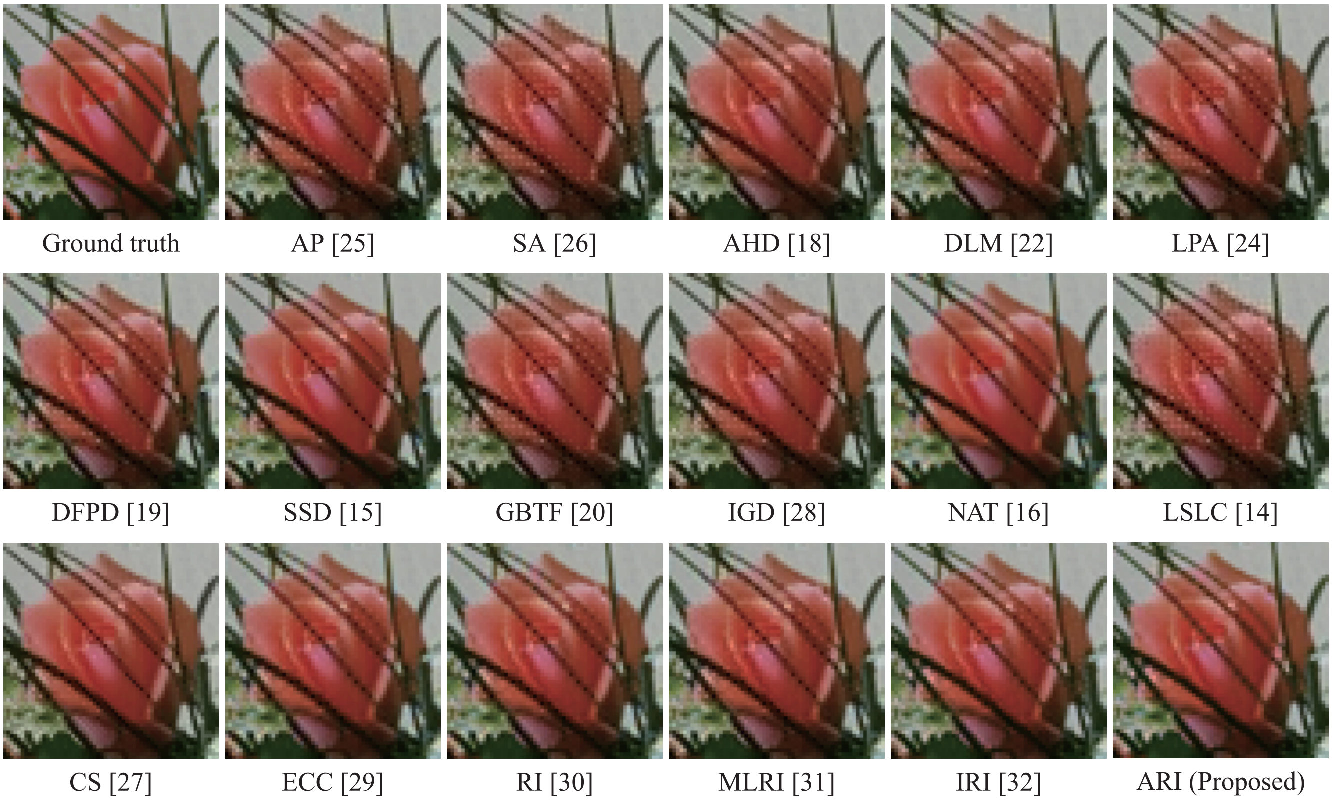

Fig. 3. Visual comparison of the demosaicking results.

Table 1. The average PSNR and CPSNR performances for the standard IMAX and Kodak datasets.

Algorithm	IMAX				Kodak				IMAX+Kodak			
	PSNR			CPSNR	PSNR			CPSNR	PSNR			CPSNR
	R	G	B		R	G	B		R	G	B	
AP [25]	32.91	35.15	32.37	33.27	39.81	43.17	39.64	40.59	35.67	38.36	35.28	36.20
SA [26]	32.73	34.73	32.10	32.98	39.80	43.31	39.50	40.54	35.56	38.16	35.06	36.01
AHD [18]	33.00	36.98	32.16	33.49	38.81	40.84	38.42	39.22	35.32	38.52	34.66	35.78
DLM [22]	34.04	38.01	33.05	34.48	41.17	43.94	40.51	41.60	36.89	40.38	36.03	37.33
LPA [24]	34.36	37.88	33.30	34.72	41.66	44.46	41.00	42.12	37.28	40.51	36.38	37.68
DFPD [19]	33.80	37.21	33.00	34.27	40.26	42.54	39.86	40.72	36.39	39.34	35.74	36.85
SSD [15]	35.02	38.27	33.80	35.23	38.83	40.51	39.08	39.40	36.54	39.16	35.91	36.90
GBTF [20]	33.48	36.59	32.71	33.89	41.71	**44.85**	41.01	42.21	36.77	39.89	36.03	37.22
IGD [28]	34.33	37.38	33.46	34.70	**41.72**	**44.85**	**41.10**	**42.26**	37.29	40.37	36.51	37.72
NAT [16]	36.31	39.82	34.50	36.27	38.30	40.49	37.94	38.77	37.11	40.08	35.87	37.27
LSLC [14]	32.31	35.09	31.93	32.85	40.36	43.85	39.89	41.03	35.53	38.59	35.11	36.12
CS [27]	35.56	38.84	34.58	35.92	41.01	44.17	40.12	41.43	37.74	40.97	36.80	38.12
ECC [29]	36.67	39.99	35.31	36.78	39.87	42.17	39.00	40.14	37.95	40.86	36.79	38.12
RI [30]	36.07	39.99	35.35	36.48	39.64	42.17	38.87	39.99	37.50	40.86	36.76	37.88
MLRI [31]	36.35	39.90	35.36	36.62	40.59	42.97	39.86	40.94	38.04	41.13	37.16	38.35
IRI [32]	36.62	40.28	35.79	36.98	40.27	43.48	39.72	40.85	38.08	41.56	37.36	38.53
Proposed	**37.41**	**40.72**	**36.05**	**37.52**	40.81	43.66	40.21	41.31	**38.77**	**41.90**	**37.72**	**39.03**

Acknowledgment This work was partly supported by Strategic Information and Communications R&D Promotion Programme (SCOPE, No.141203024) from Ministry of Internal Affairs and Communications (MIC).

5. REFERENCES

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

[2] B. K. Gunturk, J. Glotzbach, Y. Altunbasak, R. W. Schafer, and R. M. Mersereau, "Demosaicking:color filter array interpolation," *IEEE Signal Processing Magazine*, vol. 22, pp. 44–54, 2005.

[3] X. Li, B. Gunturk, and L. Zhang, "Image demosaicing: a systematic survey," *Proc. of SPIE*, vol. 6822, pp. 68221J–1–15, 2008.

[4] D. Menon and G. Calvagno, "Color image demosaicking: An overview," *Signal Processing: Image Communications*, vol. 26, no. 8, pp. 518–533, 2011.

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

[6] A. Hore and D. Ziou, "Demosaicing: image reconstruction from color CCD samples," *IEEE Trans. on Image Processing*, vol. 8, no. 9, pp. 1221–1228, 1999.

[7] R. Lukac and K. N. Plataniotis, "Normalized color-ratio modeling for CFA interpolation," *IEEE Trans. on Consumer Electronics*, vol. 50, no. 2, pp. 737–745, 2004.

[8] D. Cok, "Signal processing method and apparatus for producing interpolated chrominance values in a sampled color image signal," *U.S. Patent 4642678*, 1987.

[9] R. Hibbard, "Apparatus and method for adaptively interpolating a full color image utilizing luminance gradients," *U.S. Patent 5382976*, 1995.

[10] J. Hamilton and J. Adams, "Adaptive color plan interpolation in single sensor color electronic camera," *U.S. Patent 5629734*, 1997.

[11] D. Alleysson, S. Süsstrunk, and J. Hérault, "Linear demosaicking inspired by the human visual system," *IEEE Trans. on Image Processing*, vol. 14, no. 4, pp. 439–449, 2005.

[12] E. Dubois, "Frequency-domain methods for demosaicking of Bayer-sampled color images," *IEEE Signal Processing Letters*, vol. 12, no. 12, pp. 847–850, 2005.

[13] N. X. Lian, L. Chang, Y. P. Tan, and V. Zagorodnov, "Adaptive filtering for color filter array demosaicking," *IEEE Trans. on Image Processing*, vol. 16, no. 10, pp. 2515–2525, 2007.

[14] B. Leung, G. Jeon, and E. Dubois, "Least-squares luma–chroma demultiplexing algorithm for Bayer demosaicking," *IEEE Trans. on Image Processing*, vol. 20, no. 7, pp. 1885–1894, 2011.

[15] A. Buades, B. Coll, J. M. Morel, and C. Sbert, "Self-similarity driven color demosaicking," *IEEE Trans. on Image Processing*, vol. 18, no. 6, pp. 1192–1202, 2009.

[16] L. Zhang, X. Wu, A. Buades, and X. Li, "Color demosaicking by local directional interpolation and nonlocal adaptive thresholding," *Journal of Electronic Imaging*, vol. 20, no. 2, pp. 023016–1–16, 2011.

[17] J. Duran and A. Buades, "Self-similarity and spectral correlation adaptive algorithm for color demosaicking," *IEEE Trans. on Image Processing*, vol. 23, no. 9, pp. 4031–4040, 2014.

[18] K. Hirakawa and T. W. Parks, "Adaptive homogeneity-directed demosaicking algorithm," *IEEE Trans. on Image Processing*, vol. 14, no. 3, pp. 360–369, 2005.

[19] D. Menon, S. Andriani, and G. Calvagno, "Demosaicing with directional filtering and a posteriori decision," *IEEE Trans. on Image Processing*, vol. 16, no. 1, pp. 132–141, 2007.

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

[21] I. Pekkucuksen and Y. Altunbasak, "Multiscale gradients-based color filter array interpolation," *IEEE Trans. on Image Processing*, vol. 22, no. 1, pp. 157–165, 2013.

[22] L. Zhang and X. Wu, "Color demosaicking via directional linear minimum mean square-error estimation," *IEEE Trans. on Image Processing*, vol. 14, no. 12, pp. 2167–2178, 2005.

[23] K. H. Chung and Y. H. Chan, "Color demosaicing using variance of color differences," *IEEE Trans. on Image Processing*, vol. 15, no. 10, pp. 2944–2955, 2006.

[24] D. Paliy, V. Katkovnik, R. Bilcu, S. Alenius, and K. Egiazarian, "Spatially adaptive color filter array interpolation for noiseless and noisy data," *Int. Journal of Imaging Systems and Technology*, vol. 17, no. 3, pp. 105–122, 2007.

[25] B. K. Gunturk, Y. Altunbasak, and R. M. Mersereau, "Color plane interpolation using alternating projections," *IEEE Trans. on Image Processing*, vol. 11, no. 9, pp. 997–1013, 2002.

[26] X. Li, "Demosaicing by successive approximation," *IEEE Trans. on Image Processing*, vol. 14, no. 3, pp. 370–379, 2005.

[27] P. Getreuer, "Image demosaicking with contour stencils," *Image Processing On Line*, vol. 2, pp. 22–34, 2012.

[28] K. H. Chung and Y. H. Chan, "Low-complexity color demosaicing algorithm based on integrated gradients," *Journal of Electronic Imaging*, vol. 19, no. 2, pp. 021104–1–15, 2010.

[29] S. P. Jaiswal, O. C. Au, V. Jakhetiya, Y. Yuan, and H. Yang, "Exploitation of inter-color correlation for color image demosaicking," *Proc. of IEEE Int. Conf. on Image Processing (ICIP)*, pp. 1812–1816, 2014.

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

[31] D. Kiku, Y. Monno, M. Tanaka, and M. Okutomi, "Minimized-Laplacian residual interpolation for color image demosaicking," *Proc. of SPIE*, vol. 9023, pp. 90230L–1–8, 2014.

[32] W. Ye and K. K. Ma, "Image demosaicing by using iterative residual interpolation," *Proc. of IEEE Int. Conf. on Image Processing (ICIP)*, pp. 1862–1866, 2014.

[33] K. He, J. Sun, and X. Tang, "Guided image filtering," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 35, no. 6, pp. 1397–1409, 2013.

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

[35] Y. Monno, D. Kiku, S. Kikuchi, M. Tanaka, and M. Okutomi, "Multispectral demosaicking with novel guide image generation and residual interpolation," *Proc. of IEEE Int. Conf. on Image Processing (ICIP)*, pp. 645–649, 2014.