J Inf Process Syst, Vol.16, No.6, pp.1479~1494, December 2020
https://doi.org/10.3745/JIPS.02.0148

ISSN 1976-913X (Print)
ISSN 2092-805X (Electronic)

JOURNAL OF INFORMATION PROCESSING SYSTEMS JIPS

Directional Interpolation Based on Improved Adaptive Residual Interpolation for Image Demosaicking

Chenbo Liu*

Abstract

As an important part of image processing, image demosaicking has been widely researched. It is especially necessary to propose an efficient interpolation algorithm with good visual quality and performance. To improve the limitations of residual interpolation (RI), based on RI algorithm, minimalized-Laplacian RI (MLRI), and iterative RI (IRI), this paper focuses on adaptive RI (ARI) and proposes an improved ARI (IARI) algorithm which obtains more distinct R, G, and B colors in the images. The proposed scheme fully considers the brightness information and edge information of the image. Since the ARI algorithm is not completely adaptive, IARI algorithm executes ARI algorithm twice on R and B components according to the directional difference, which surely achieves an adaptive algorithm for all color components. Experimental results show that the improved method has better performance than other four existing methods both in subjective assessment and objective assessment, especially in the complex edge area and color brightness recovery.

Keywords

Adaptive Residual Interpolation (ARI), Directional Difference, Image Demosaicking, Iterative Residual Interpolation (IRI), Minimized-Laplacian Residual Interpolation (MLRI), Residual Interpolation (RI)

1. Introduction

To save manufacturing costs and reduce the size of the device, most digital imaging devices use monochrome sensors that can only obtain the change of single color. Therefore, an algorithm must be used to estimate the loss of color to obtain a full-color image. This process is called demosaicking, which belongs to image interpolation. Image demosaicking is the key to imaging process, so it is especially necessary to improve the image quality by improving performance of the algorithm [1].

Since the image sensor captures the change of single color, Bayer [2] covered the photosensitive unit with an optical filter called color filter array (CFA). Currently, Bayer pattern CFA becomes the most commonly used model, and its schematic diagram is 5×5 pixels, which is shown in Fig. 1, where R, G, and B respectively represent color components obtained after passing through a filter of red, green, and blue. Their positions are marked as L_R, L_G, and L_B, respectively.

Based on Bayer pattern CFA algorithm, the earliest and most widely used interpolation algorithm was bilinear interpolation [3], which decomposes CFA into three mutually independent images and processes them independently. This algorithm is not complicated, which can suppress image overlapping

Manuscript received November 12, 2019; first revision April 9, 2020; accepted April 22, 2020.
Corresponding Author: Chenbo Liu (lcbedu@163.com)
* College of Engineering, Qufu Normal University (Rizhao Campus), Shandong, China (lcbedu@163.com)

effectively. However, in the area with complex edges, color artifacts and serrated effects are produced by bilinear interpolation [3], therefore, the quality of the image is usually imperfect. To improve this problem, researchers began to study the correlation of spectrum among different color components. Gunturk et al. [4] found that the correlation among different color components was mainly determined by smoothness of chromatic aberration, which means, the smoother the color plane, the better the image interpolation effect. This theory led to some algorithms based on color interpolation (CI) [5,6] being proposed. Zhang and Xu [5] proposed a directional linear minimum mean square-error (DLMMSE) estimation algorithm, which estimates the missing G component adaptively in different directions of the image. Although the DLMMSE algorithm obtains optimal color-difference evaluation at each individual pixel location, it ignores the additional information that may be provided by neighboring pixel locations. To obtain such additional information, Pkkucuksen and Altunbasak [6] proposed a gradient based threshold free (GBTF) algorithm, which uses a color-difference gradient instead of a color-difference variance to estimate the missing G component. The GBTF algorithm not only obtains additional information of neighboring pixel locations, but also eliminates the hard rules of DLMMSE, so the threshold setting is no longer needed. The GBTF algorithm assumes that the image has hyperspectral correlation. However, in natural environment, spectral correlation is usually weaker in the edge regions of the image than in the smooth regions.

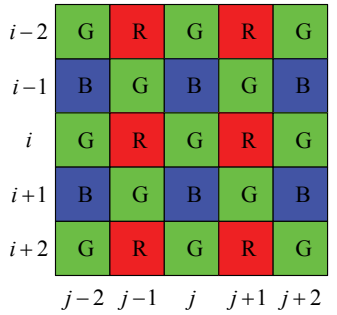

Fig. 1. Bayer pattern color filter array (CFA).

In recent years, Kiku et al. [7] proposed the residual interpolation (RI), which uses a smoother residual plane than color plane for interpolation, and achieves a remarkable image restoration effect. Based on RI method, Kiku et al. [8] found that the smaller the Laplacian energy (L_E) of the image, the better the image interpolation effect. Therefore, they proposed the minimized-Laplacian RI (MLRI) to minimize the value of L_E. However, in these algorithms [7,8], the prediction error generated during the generation of G component affects the generation of R and B components. To improve this problem, Ye and Ma [9] proposed iterative RI (IRI) that tested the test datasets iteratively to reduce the estimation error. Kim and Jeong [10] combined the estimated pixel values with the inverse gradient weights in four directions, so the prediction error can be reduced. Yu et al. [11] weakened the effects of gradient weights in horizontal (H) and vertical (V) directions on the prediction error. Monno et al. [12] used the iterative thought of IRI to combine RI with MLRI, and then proposed adaptive RI (ARI) to generate G component, which improved the quality of image restoration. However, in the ARI algorithm, a non-adaptive MLRI algorithm is still adopted to obtain R and B components. Recently, Ni et al. [13] proposed a progressive collaborative representation (PCR) framework, which can incorporate any existing color image

demosaicking method and further improve its demosaicking performance. Sun et al. [14] proposed a hybrid demosaicking algorithm based on fuzzy edge strength and RI method, which has better performance in terms of structural similarity and visual comparison.

To solve the problem of ARI algorithm, the adaptability of the algorithm is not only applicable to G component, but also to R and B components. In addition, the interpolation of R and B components includes not only H and V directions but also the diagonal direction. Meanwhile, it is more difficult to use the ARI algorithm directly for both types of interpolation directions simultaneously. According to disadvantages mentioned above, this paper proposes a solution. This paper takes the RI algorithm as research object. The main research contents include: understanding the principle of obtaining smooth residual plane by RI, MLRI, and IRI algorithms; reproducing ARI algorithm; improving the interpolation process of R and B components in the improved ARI (IARI) algorithm; testing and evaluating the image restoration effects of the five algorithms.

The rest of this paper is organized as follows. Section 2 starts with a brief review of RI algorithm basis and its derivative algorithm, and then ARI algorithm is described. The proposed method is designed in Section 3. Experimental results of the proposed method are presented in Section 4. Conclusions and remarks on possible further work are given finally in Section 5.

2. Related Work

2.1 Residual Interpolation Algorithm Basis and Its Derivative Algorithm

The basic flow of RI algorithm is described in detail in [7]. The important details in RI algorithm include the interpolation of G component by the GBTF interpolation algorithm [6], and the guided filter (GF) of the image $\tilde{\boldsymbol{G}}$ generated by G component after interpolation [15], where the GBTF interpolation algorithm is implemented on the basis of the Hamilton interpolation formula [15]. From [16], in fact, in the RI algorithm, the cost function minimizes the value of the residual. Therefore, the essence of the RI algorithm is to obtain the smooth residual plane by minimizing the residual. After that, based on the RI algorithm, Kiku et al. [8] proposed the MLRI algorithm according to the relation that the image smoothness index L_E is negatively correlated with the RI effect. The basic flow of MLRI algorithm is described in detail in [8]. Moreover, it can be seen that in the MLRI algorithm, the cost function minimizes the image smoothness index L_E, which is a measure of image smoothness. Therefore, the essence of the MLRI algorithm is to obtain the smooth residual plane by minimizing the image smoothness.

Later, Ye and Ma [9] proposed the IRI algorithm, which makes the residual plane smooth by iteration. The basic flow of IRI algorithm is described in detail in [9]. It can be seen that the residual plane becomes smoother and smoother during the iterative repetition. Therefore, the essence of the IRI algorithm is to minimize the residual by iterating RI's minimized residual to obtain the smooth residual plane. Since the local window size gradually increases during iterative process, it is impossible to judge the iterative coefficient [17]. Hence, Ye and Ma [9,17] determined the local window size which is an important parameter of GF and has a huge effect on the performance of the algorithm, thereby determining the optimal iterative coefficient.

2.2 Adaptive Residual Interpolation Algorithm

The different principles of RI, MLRI, and IRI algorithms for obtaining smooth residual planes have advantages in images with different characteristics. Based on RI, MLRI, and IRI algorithms, the optimal algorithm at different pixel locations is adopted adaptively in ARI algorithm to obtain better G component interpolation images [12].

Since the calculation of G component at L_R and L_B is the same, this paper only introduces the calculation of G component at L_R. The following part describes three steps in G component interpolation process of the ARI algorithm [12]. Since the interpolation and selection process of the iterative coefficients in H and V directions are the same, this paper only introduces the algorithm and selection process in the H direction.

Step 1: To obtain a linear G component, the RI and MLRI algorithms are utilized iteratively in H and V directions to obtain a linear G component. For each interpolated result, a set of directional interpolated G component images will be generated, and each of them corresponds to one iteration [12].

Step 2: For the interpolation results in each direction, the optimal iteration number is adaptively selected from the set of directional interpolated G component images at each pixel position, based on iterative rule proposed in [17]. The iterative criteria $c^{\mathrm{H}}_{(i,j),k}$ at the pixel location (i,j) in the kth iteration can be expressed as [12]:

$$c^{\mathrm{H}}_{(i,j),k} = (d^{\mathrm{H}}_{(i,j),k})^m \cdot (\delta d^{\mathrm{H}}_{(i,j),k})^n \tag{1}$$

where the superscript $(\cdot)^{\mathrm{H}}$ represents the H direction, and the subscript $(\cdot)_{(i,j),k}$ represents the kth interpolation at (i,j). Meanwhile, $d^{\mathrm{H}}_{(i,j),k}$ represents the difference between the estimated pixel value and the previous interpolation results, and $\delta d^{\mathrm{H}}_{(i,j),k}$ represents the smoothness of the difference. They can be denoted as [12]:

$$d^{\mathrm{H}}_{(i,j),k} = \left|d^{\mathrm{GH}}_{(i,j),k}\right| + \left|d^{\mathrm{RH}}_{(i,j),k}\right|,\ \delta d^{\mathrm{H}}_{(i,j),k} = \left|d^{\mathrm{GH}}_{(i,j-1),k} - d^{\mathrm{GH}}_{(i,j+1),k}\right| + \left|d^{\mathrm{RH}}_{(i,j-1),k} - d^{\mathrm{RH}}_{(i,j+1),k}\right| \tag{2}$$

where the superscripts $(\cdot)^{\mathrm{GH}}$ and $(\cdot)^{\mathrm{RH}}$ represent the residual calculation process of R and B components in H direction, respectively. In addition, the parameters (m,n) are set to the optimal solution $(2,1)$ obtained in [17]. The optimal iteration coefficient k_{best} for each pixel location can be adaptively expressed as [12]:

$$k_{\mathrm{best}} = \arg\min_k g(c^{\mathrm{H}}_{(i,j),k}) \tag{3}$$

where the function $g(\cdot)$ represents spatial Gaussian smoothing filter with the standard deviation σ=2 [12]. When k_{best} is determined, according to the interpolation results of Step 1, the optimal results can be obtained by interpolating RI or MLRI algorithm corresponding to each pixel position in the H or V direction.

Step 3: Weighted average at each L_R can adaptively combine the results of RI and MLRI algorithms interpolated in different directions, and the final interpolated image $\tilde{\boldsymbol{G}}$ will be obtained.

3. Improved Adaptive Residual Interpolation Algorithm

The adaptability of the conventional ARI algorithm is only applied to the interpolation process of G component, and the interpolation of R and B components still uses the conventional MLRI algorithm. When the conventional ARI algorithm performs iterative interpolation on G component, it only contains interpolation directions of H and V. However, the interpolation of R and B components includes not only H and V directions, but also the diagonal direction [12]. Moreover, it is difficult to use the ARI algorithm directly for two types of interpolation directions simultaneously. To solve this problem, the interpolation process of R and B components is divided into two categories according to the direction types [12]: (i) H and V directions and (ii) diagonal direction. The ARI algorithm is performed once in two categories, respectively. The processing of B component is the same as R component, and then the interpolation process of R component includes the following two steps:

Step 1: The ARI algorithm is used to obtain R component in each of two directions of upper left to lower right (labeled as D_1 direction) and upper right to lower left (marked as D_2 direction) at each L_B.

Step 2: The ARI algorithm is used at each of the L_G directions in H and V directions to obtain R component.

Since the ARI algorithm is used along H and V directions in Step 2, which is the same as the interpolation method in Section 2.2, and then only the diagonal ARI algorithm of Step 1 is introduced.

Step 1 iteratively performs two kinds of algorithms (RI and MLRI algorithms) in two directions (D_1 and D_2 directions), and the RI algorithm flow of R component in D_1 direction is shown in Fig. 2, where the dotted line direction represents D_1 direction. The calculation process of MLRI algorithm in D_1 direction and RI and MLRI algorithms in D_2 direction are performed in the same way.

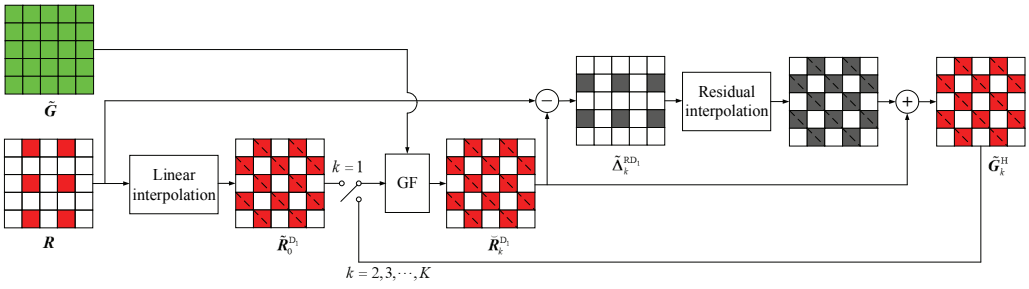

Fig. 2. Interpolation process of R component in the direction of D_1.

Firstly, an initial linear interpolation of R component in D_1 direction is performed at L_B, and the interpolation results can be represented as:

$$\tilde{R}_{(i,j),0}^{D_1} = \frac{R_{(i-1,j-1)}^{D_1} + R_{(i+1,j+1)}^{D_1}}{2} \tag{4}$$

where $\tilde{R}_{(i,j)}$ represents R component of image $\tilde{\boldsymbol{R}}$ at (i,j), $R_{(i-1,j-1)}$ and $R_{(i+1,j+1)}$ represent R component of image $\boldsymbol{R}$ at $(i-1,j-1)$ and $(i+1,j+1)$, respectively. Meanwhile, the superscript $(\cdot)^{D_1}$ represents that the interpolation process is in D_1 direction, and the subscript $(\cdot)_{(i,j),0}$ represents the initial interpolation at (i,j).

Secondly, at the *k*th iteration, an estimated image $\breve{\boldsymbol{R}}$ is generated based on the linear transformation of the previous interpolation results, and R component of the image at (i,j) can be expressed as:

$$\breve{R}_{(i,j),k}^{\mathrm{D}_1} = a_{(i,j),k}^{\mathrm{R}} \tilde{R}_{(i,j),k-1}^{\mathrm{D}_1} + b_{(i,j),k}^{\mathrm{R}} \tag{5}$$

where $(a^{\mathrm{R}}, b^{\mathrm{R}})$ represents the linear coefficient of R component during the estimation process. Then, the residual component $\tilde{\Delta}$ is obtained as:

$$\tilde{\Delta}_{(i,j),k}^{\mathrm{RD}_1} = R_{(i,j)} - \breve{R}_{(i,j),k}^{\mathrm{D}_1} \tag{6}$$

where the superscript $(\cdot)^{\mathrm{RD}_1}$ represents the residual calculation process of R component in D_1 direction. The residual value is linearly interpolated in D_1 direction, and the result can be expressed as:

$$\tilde{\Delta}_{(i,j),k}^{\mathrm{RH}} = \frac{\tilde{\Delta}_{(i-1,j-1),k}^{\mathrm{RH}} + \tilde{\Delta}_{(i+1,j+1),k}^{\mathrm{RH}}}{2} \tag{7}$$

Finally, to obtain the results of the *k*th iteration interpolation in D_1 direction, the residual of the linear interpolation is added to $\breve{\boldsymbol{R}}$, which can be expressed as:

$$\tilde{R}_{(i,j),k}^{\mathrm{D}_1} = \tilde{\Delta}_{(i,j),k}^{\mathrm{D}_1} + \breve{R}_{(i,j),k}^{\mathrm{D}_1} \tag{8}$$

After the *k*th iteration, according to the steps of the ARI algorithm, the optimal iteration coefficients of the four algorithms need to be determined. Since there is only R component at this time, Eq. (2) can be rewritten as:

$$d_{(i,j),k}^{\mathrm{D}_1} = \left| d_{(i,j),k}^{\mathrm{RD}_1} \right|,\ \delta d_{(i,j),k}^{\mathrm{D}_1} = \left| d_{(i-1,j-1),k}^{\mathrm{RD}_1} - d_{(i+1,j+1),k}^{\mathrm{RD}_1} \right| \tag{9}$$

At this point, the optimal iteration coefficient k_{best} obtained by Eq. (3) becomes:

$$k_{\mathrm{best}} = \arg\min_{k} g(c_{(i,j),k}^{\mathrm{D}_1}) \tag{10}$$

where $c_{(i,j),k}^{\mathrm{D}_1}$ can be expressed as:

$$c_{(i,j),k}^{\mathrm{D}_1} = (d_{(i,j),k}^{\mathrm{D}_1})^m \cdot (\delta d_{(i,j),k}^{\mathrm{D}_1})^n \tag{11}$$

where the parameters (m,n) are still set to $(2,1)$.

After obtaining the optimal iterative coefficients, $\tilde{R}_{(i,j),k_{\mathrm{best}}}^{\mathrm{D}_1}$ needs to be adaptively combined. In this step, the direction interpolation results of the RI and MLRI algorithms are combined by weighted average:

$$\tilde{R}_{(i,j)} = \frac{\omega_{(i,j)}^{\mathrm{D}_1,\mathrm{RI}} \tilde{G}_{(i,j)}^{\mathrm{D}_1,\mathrm{RI}} + \omega_{(i,j)}^{\mathrm{D}_1,\mathrm{MLRI}} \tilde{G}_{(i,j)}^{\mathrm{D}_1,\mathrm{MLRI}} + \omega_{(i,j)}^{\mathrm{D}_2,\mathrm{RI}} \tilde{G}_{(i,j)}^{\mathrm{D}_2,\mathrm{RI}} + \omega_{(i,j)}^{\mathrm{D}_2,\mathrm{MLRI}} \tilde{G}_{(i,j)}^{\mathrm{D}_2,\mathrm{MLRI}}}{\omega_{(i,j)}^{\mathrm{D}_1,\mathrm{RI}} + \omega_{(i,j)}^{\mathrm{D}_1,\mathrm{MLRI}} + \omega_{(i,j)}^{\mathrm{D}_2,\mathrm{RI}} + \omega_{(i,j)}^{\mathrm{D}_2,\mathrm{MLRI}}} \tag{12}$$

where the superscripts $(\cdot)^{\mathrm{D}_1}$ and $(\cdot)^{\mathrm{D}_2}$ represent the D_1 direction and the D_2 direction, respectively. The weights $\omega_{(i,j)}^{\mathrm{D}_1,\mathrm{RI}}$, $\omega_{(i,j)}^{\mathrm{D}_1,\mathrm{MLRI}}$, $\omega_{(i,j)}^{\mathrm{D}_2,\mathrm{RI}}$, and $\omega_{(i,j)}^{\mathrm{D}_2,\mathrm{MLRI}}$ in Eq. (12) can be expressed as:

$$\omega_{(i,j)}^{D_1,RI}=\frac{1}{g(c_{(i,j)}^{D_1,RI})},\ \omega_{(i,j)}^{D_1,MLRI}=\frac{1}{g(c_{(i,j)}^{D_1,ML})},\ \omega_{(i,j)}^{D_2,RI}=\frac{1}{g(c_{(i,j)}^{D_2,RI})},\ \omega_{(i,j)}^{D_2,MLRI}=\frac{1}{g(c_{(i,j)}^{D_2,ML})} \tag{13}$$

4. Experimental Results and Performance Analysis

Obtaining high-quality restored images is the ultimate goal of demosaicking, so it is necessary to evaluate the quality of the interpolated images objectively as well as subjectively. This paper first introduces the test image and experimental environment, and then evaluates the image quality of the five algorithms, that is, RI, MLRI, IRI, ARI, and IARI, objectively and subjectively. Finally, the average interpolation calculation time of the five algorithms is utilized to reflect the complexity of the algorithm.

4.1 Test Image and Experimental Environment

To evaluate the quality of five algorithms, this paper selects two types test images from IMAX and Kodak datasets. The representative survey in [18] points out that these two datasets are reliable samples of the demosaick image quality test. The IMAX dataset contains 18 images of 500×500 pixels in TIF format, as shown in Fig. 3, while the Kodak dataset contains 24 PNG images with size of 768×512 pixels, as shown in Fig. 4.

The software used in all the simulation experiments in this paper is MATLAB R2016a, and the operating environment is a notebook computer with 2.40 GHz Intel Core i7-5500U CPU.

Fig. 3. Eighteen test images in IMAX dataset.

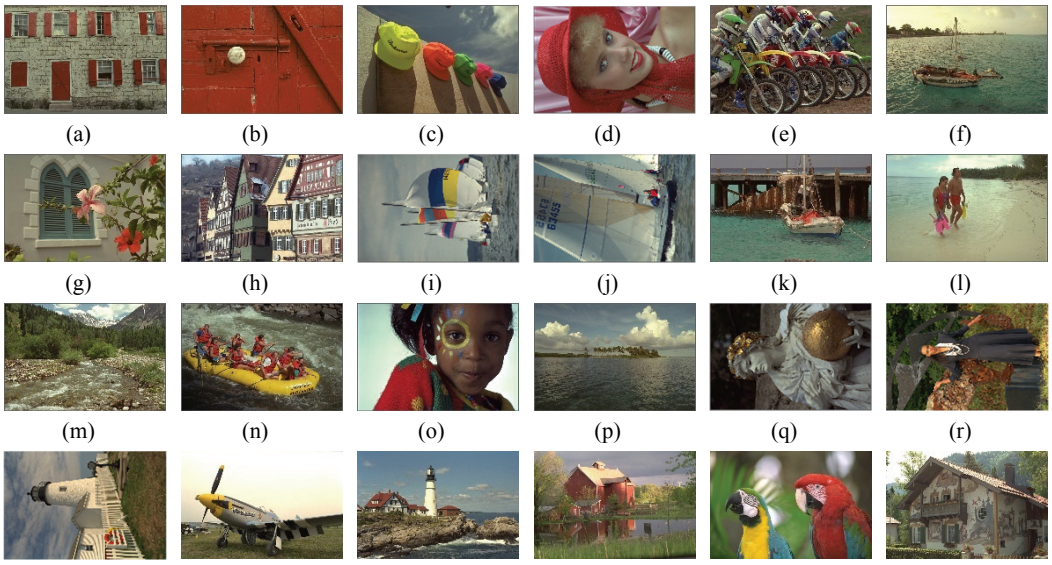

Fig. 4. Twenty-four test images in Kodak dataset.

4.2 Comparison of Objective Quality of Images

In this section, the peak signal-to-noise ratio (PSNR) and the composite PSNR (CPSNR) are used to test the performance of the demosaicking algorithm.

As an important indicator of image quality, PSNR is denoted by Q_{PSNR} in this paper, which can be expressed as [11]:

$$Q_{\text{PSNR}} = 10\log_{10}\left[\frac{255^2}{\frac{1}{MN}\sum_{i=1}^{M}\sum_{j=1}^{N}[I_{\text{in}}(i,j)-I_{\text{out}}(i,j)]^2}\right] \text{(dB)} \tag{14}$$

where I_{in} is the input image and I_{out} is the output image, and $M \times N$ is the size of the image. i and j represent the specific location of a pixel in the image. Since the PSNR is only for the evaluation quality of one of the R, G, and B colors, the overall evaluation of the image quality is not comprehensive. Therefore, this paper also introduces CPSNR, which is denoted by Q_{CPSNR} in this paper, to evaluate the overall performance of the interpolation algorithm. The Q_{CPSNR} can be expressed as [11]:

$$Q_{\text{CPSNR}} = 10\log_{10}\left[\frac{255^2}{\frac{1}{3MN}\sum_{k=1}^{3}\sum_{i=1}^{M}\sum_{j=1}^{N}[I_{\text{in}}(i,j,k)-I_{\text{out}}(i,j,k)]^2}\right] \text{(dB)} \tag{15}$$

where $k = 1,2,3$ represents the color plane of R, G, and B components, respectively.

Table 1 shows the comparison of the Q_{PSNR} of G, R, and B components after processing the 42 images in IMAX and Kodak datasets among five methods mentioned above.

From Table 1, it can be seen that the average Q_{PSNR} of each color component of the IARI algorithm is improved, compared with the other algorithms, in terms of the most images. Specifically, the average Q_{PSNR} of G component of the IARI and ARI algorithms is 1.25 dB, 0.81 dB, and 0.62 dB higher than that of the RI, MLRI, and IRI algorithms, respectively. In addition, the average Q_{PSNR} of R component of the IARI is 1.47 dB, 0.74 dB, 0.77 dB, and 0.12 dB higher than that of the RI, MLRI, IRI, and ARI algorithms, respectively. The average Q_{PSNR} of B component by the method of the IARI is 1.22 dB, 0.69 dB, 0.37 dB, and 0.29 dB higher than that of the RI, MLRI, IRI, and ARI algorithms, respectively. To improve readability, the graphs which is simulated by MATLAB R2016a are used to further illustrate, as shown in Figs. 5–7. These three graphs show the comparison of Q_{PSNR} of G, R, and B components among five algorithms, respectively. For convenience, Fig. 3(a)–(r) are numbered 1 to 18 in sequence, and Fig. 4(a)–(x) are numbered 19 to 42 in sequence. The image numbers are recorded as parameter α.

From Fig. 5, it can be seen that the curves of ARI and IARI algorithms overlap completely, that is, the IARI and ARI algorithms have the same effect on the interpolation of G component. From Figs. 6 and 7, it can be seen that in most cases, the curve of IARI algorithm is located above other curves, that is, for most images, the Q_{PSNR} of R and G components of the IARI algorithm are better than that of other four algorithms. Table 2 shows the comparison of Q_{CPSNR}, and the bolded data in table is the optimal results for each image. Moreover, to improve readability, Fig. 8 is used to illustrate the comparison of Q_{CPSNR} among five algorithms.

Table 1. Comparison of Q_{PSNR} (dB) of R, G, and B components among five algorithms

Test images	Q_{PSNR} (dB) of G component					Q_{PSNR} (dB) of R component					Q_{PSNR} (dB) of B component				
	RI [7]	MLRI [8]	IRI [9]	ARI [12]	IARI	RI [7]	MLRI [8]	IRI [9]	ARI [12]	IARI	RI [7]	MLRI [8]	IRI [9]	ARI [12]	IARI
Fig. 3(a)	32.37	32.48	33.40	**33.61**	**33.61**	29.16	29.21	**31.22**	30.01	30.10	26.98	26.93	**30.52**	27.65	27.81
Fig. 3(b)	39.41	39.38	39.31	**39.44**	**39.44**	34.48	34.67	34.25	**35.01**	34.97	33.21	33.30	**33.60**	33.28	33.46
Fig. 3(c)	36.75	36.99	37.02	**37.65**	**37.65**	34.04	34.18	34.06	34.86	**35.18**	31.74	31.97	**33.15**	32.76	33.05
Fig. 3(d)	42.14	41.85	41.80	**42.26**	**42.26**	38.30	38.31	37.31	38.27	**38.55**	35.53	35.35	35.50	35.60	**35.71**
Fig. 3(e)	37.86	38.36	38.92	**39.15**	**39.15**	36.80	36.90	36.33	38.35	**38.50**	30.73	30.85	33.28	32.30	**32.40**
Fig. 3(f)	42.16	41.83	41.57	**42.51**	**42.51**	39.08	39.06	37.76	**40.62**	40.35	35.89	35.93	35.84	**37.36**	37.11
Fig. 3(g)	38.77	39.39	41.28	**42.42**	**42.42**	37.09	37.54	37.61	39.54	**39.79**	35.62	36.14	36.91	38.26	**39.24**
Fig. 3(h)	41.37	41.68	42.49	**42.88**	**42.88**	34.29	34.14	37.65	**37.97**	37.85	38.07	38.25	38.90	39.13	**39.24**
Fig. 3(i)	41.62	41.53	41.56	**42.27**	**42.27**	33.25	34.20	35.37	**36.49**	36.37	36.49	36.51	36.48	36.94	**37.14**
Fig. 3(j)	42.07	42.37	41.13	**42.54**	**42.54**	36.73	37.63	36.66	**38.75**	38.73	37.34	37.60	36.85	37.82	**38.02**
Fig. 3(k)	42.03	41.99	42.06	**42.12**	**42.12**	37.87	39.02	37.60	39.59	**39.61**	39.38	39.39	**39.99**	39.46	39.83
Fig. 3(l)	42.24	42.35	42.50	**42.79**	**42.79**	40.26	40.26	39.57	40.55	**40.62**	37.65	37.75	37.89	38.11	**38.48**
Fig. 3(m)	45.10	44.91	45.12	**45.24**	**45.24**	42.63	42.23	41.30	42.43	**42.68**	37.13	37.65	37.33	37.65	**38.17**
Fig. 3(n)	42.95	42.86	42.99	**43.14**	**43.14**	39.38	39.34	38.41	39.40	**39.44**	36.64	36.42	36.39	36.78	**36.95**
Fig. 3(o)	42.67	42.57	42.71	**42.92**	**42.92**	35.78	36.92	36.66	37.56	**37.65**	39.19	39.09	38.64	39.33	**39.40**
Fig. 3(p)	35.16	35.24	34.10	**35.50**	**35.50**	34.44	34.38	32.64	35.08	**35.11**	35.98	35.75	34.68	36.46	**36.62**
Fig. 3(q)	37.38	37.27	37.91	**38.87**	**38.87**	30.55	31.25	32.58	**33.80**	33.33	31.78	31.53	**33.54**	33.29	33.10
Fig. 3(r)	37.68	37.71	37.78	**37.99**	**37.99**	34.18	35.00	34.21	35.01	**35.31**	36.97	36.16	36.13	36.79	**36.93**

Table 1. (Continued)

Test images	Q_{PSNR} (dB) of G component					Q_{PSNR} (dB) of R component					Q_{PSNR} (dB) of B component				
	RI [7]	MLRI [8]	IRI [9]	ARI [12]	IARI	RI [7]	MLRI [8]	IRI [9]	ARI [12]	IARI	RI [7]	MLRI [8]	IRI [9]	ARI [12]	IARI
Fig. 4(a)	37.33	38.45	38.36	**41.34**	**41.34**	34.49	35.99	36.95	**37.65**	37.50	35.34	36.35	38.18	38.34	**38.49**
Fig. 4(b)	43.02	43.88	43.96	**44.65**	**44.65**	36.72	38.41	37.75	37.40	**37.47**	41.13	**41.84**	40.85	40.67	41.08
Fig. 4(c)	45.04	45.21	45.06	**45.47**	**45.47**	39.65	41.44	41.40	42.05	**42.18**	40.18	41.71	41.38	41.63	**41.88**
Fig. 4(d)	42.96	43.25	43.21	**43.29**	**43.29**	37.81	38.27	38.24	**38.38**	38.24	41.61	42.18	41.51	41.74	**42.21**
Fig. 4(e)	38.98	39.91	40.54	**40.83**	**40.83**	36.19	37.30	37.04	**37.92**	37.90	35.54	36.48	**37.10**	36.98	36.99
Fig. 4(f)	40.28	41.09	39.48	**43.31**	**43.31**	38.17	38.96	39.78	40.07	**40.23**	37.38	37.93	**40.48**	39.25	39.50
Fig. 4(g)	44.79	45.28	44.93	**45.32**	**45.32**	41.30	42.35	41.83	42.31	**42.41**	40.84	41.69	41.18	41.64	**41.75**
Fig. 4(h)	36.36	37.20	36.82	**37.81**	**37.81**	33.13	34.09	34.23	34.07	**34.45**	33.19	34.10	33.95	34.06	**34.38**
Fig. 4(i)	44.02	44.68	44.75	**44.82**	**44.82**	41.27	**42.10**	41.65	41.74	41.77	39.78	**41.09**	40.72	40.53	40.46
Fig. 4(j)	44.45	45.00	45.12	**45.48**	**45.48**	41.22	41.38	41.50	41.21	**41.57**	40.28	**41.01**	40.31	40.49	40.79
Fig. 4(k)	40.06	41.05	41.78	**41.94**	**41.94**	37.19	38.40	38.30	38.43	**38.58**	37.78	38.93	39.23	39.34	**39.56**
Fig. 4(l)	45.23	46.01	46.32	**46.42**	**46.42**	41.15	42.22	41.99	41.97	**42.49**	41.75	42.50	42.39	42.55	**42.79**
Fig. 4(m)	33.18	34.43	37.10	**37.20**	**37.20**	31.92	33.15	34.85	35.20	**35.56**	31.07	32.16	34.04	33.88	**34.06**
Fig. 4(n)	39.40	40.41	40.04	**40.89**	**40.89**	35.06	36.48	36.68	36.62	**36.94**	35.58	36.94	38.02	36.90	**39.49**
Fig. 4(o)	42.15	42.45	42.83	**43.22**	**43.22**	36.65	37.13	37.62	37.77	**37.84**	39.28	40.06	**40.34**	39.88	40.24
Fig. 4(p)	43.93	44.69	45.05	**45.46**	**45.46**	42.05	42.62	42.14	42.62	**42.94**	41.25	41.84	42.03	42.49	**42.71**
Fig. 4(q)	41.99	42.74	42.12	**43.62**	**43.62**	40.36	41.21	41.60	41.63	**41.89**	38.88	39.44	**39.80**	39.52	39.66
Fig. 4(r)	37.00	38.01	38.45	**38.90**	**38.90**	35.55	36.54	36.69	36.51	**37.02**	34.29	35.60	35.66	35.91	**36.03**
Fig. 4(s)	40.95	41.79	42.29	**43.00**	**43.00**	38.77	39.58	39.13	39.86	**40.22**	38.24	39.02	38.95	**39.73**	39.71
Fig. 4(t)	42.21	43.16	43.16	**44.16**	**44.16**	40.59	41.48	41.48	41.89	**42.19**	37.99	38.63	38.63	38.49	**39.09**
Fig. 4(u)	39.16	40.17	40.56	**41.94**	**41.94**	37.18	38.15	39.27	39.14	**39.41**	35.95	36.87	37.53	37.73	**37.89**
Fig. 4(v)	39.92	40.73	40.90	**40.91**	**40.91**	36.85	**38.19**	37.39	37.80	38.00	36.59	37.55	37.52	36.91	**37.57**
Fig. 4(w)	45.64	45.26	45.74	**45.83**	**45.83**	40.80	42.64	41.68	**42.38**	42.37	41.57	**43.34**	42.59	42.54	42.68
Fig. 4(x)	36.20	36.85	37.80	**37.88**	**37.88**	34.40	35.36	36.16	35.92	**36.19**	32.54	32.82	33.34	33.42	**33.75**
Average	40.57	41.01	41.19	**41.69**	**41.69**	37.07	37.80	37.77	38.42	**38.54**	36.77	37.30	37.62	37.70	**37.99**

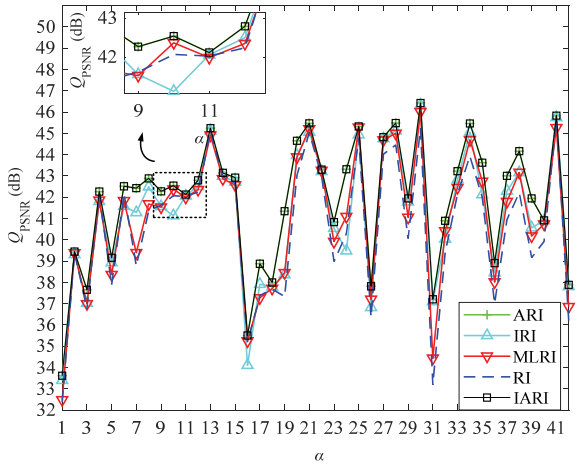

Fig. 5. Comparison of Q_{PSNR} (dB) of G component among five algorithms.

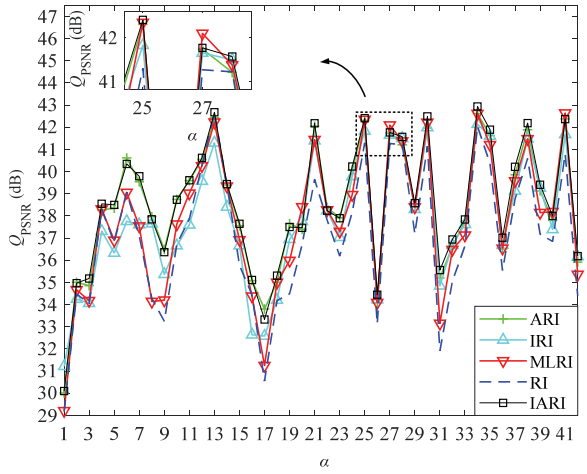

Fig. 6. Comparison of Q_{PSNR} (dB) of R component among five algorithms.

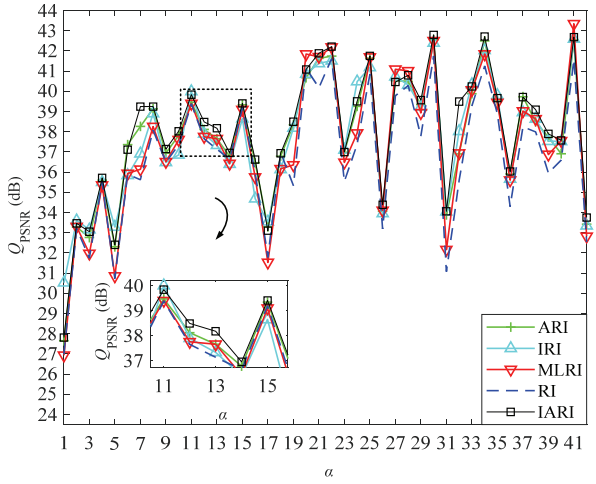

Fig. 7. Comparison of Q_{PSNR} (dB) of B component among five algorithms.

Table 2 shows that the average Q_{CPSNR} of the IARI algorithm is improved, compared with other algorithms, in terms of the most images. Specifically, the average Q_{CPSNR} of the IARI algorithm is 1.43 dB, 0.68 dB, 0.42 dB, and 0.26 dB higher than that of RI, MLRI, IRI, and ARI algorithms, respectively. From Fig. 8, it can be seen that in most cases, the curve of IARI algorithm is located above other curves, that is, for most images, the Q_{CPSNR} of the IARI algorithm is higher than that of other four algorithms.

In summary, experimental results show that the IARI algorithm is not all superior to the ARI algorithm in any case. This paper believes that the reason for these results is that in order to be consistent with the ARI algorithm of G component, the iteration parameters (m,n) in Eq. (11) are still set to $(2,1)$. However, $(2,1)$ are not necessarily the optimal parameter in the interpolation process on R and B components. In summary, experimental results show that the interpolation effect of the IARI algorithm is better than that of other four comparison algorithms.

Table 2. Comparison of Q_{CPSNR} (dB) among five algorithms

Test images	RI [7]	MLRI [8]	IRI [9]	ARI [12]	IARI
Fig. 3(a)	28.98	29.27	**31.77**	29.63	29.74
Fig. 3(b)	35.00	**35.40**	35.17	35.23	35.31
Fig. 3(c)	33.71	34.18	34.57	34.65	**34.92**
Fig. 3(d)	37.88	37.97	37.55	37.92	**38.08**
Fig. 3(e)	33.92	34.32	**35.57**	35.44	35.53
Fig. 3(f)	38.32	38.59	37.84	**39.79**	39.54
Fig. 3(g)	36.97	37.79	38.26	**39.75**	39.53
Fig. 3(h)	36.98	37.27	39.36	**39.55**	39.53
Fig. 3(i)	35.92	36.76	37.23	37.90	**37.93**
Fig. 3(j)	38.15	38.98	37.85	39.28	**39.35**
Fig. 3(k)	39.44	40.25	38.93	40.23	**40.38**
Fig. 3(l)	39.64	39.98	39.60	40.07	**40.26**
Fig. 3(m)	40.31	40.86	40.19	40.64	**41.02**
Fig. 3(n)	38.95	39.09	38.56	39.05	**39.14**
Fig. 3(o)	38.35	39.24	38.82	39.41	**39.48**
Fig. 3(p)	35.15	35.38	33.65	35.64	**35.67**
Fig. 3(q)	32.39	32.89	34.24	**34.70**	34.41
Fig. 3(r)	36.48	36.42	35.89	36.43	**36.54**
Fig. 4(a)	35.56	36.80	38.56	**38.84**	**38.84**
Fig. 4(b)	39.46	**40.78**	40.11	39.69	39.69
Fig. 4(c)	41.04	**42.99**	42.73	42.75	42.95
Fig. 4(d)	40.22	40.45	40.59	40.62	**40.67**
Fig. 4(e)	36.67	37.67	37.98	38.29	**38.30**
Fig. 4(f)	38.45	39.14	**40.96**	40.55	40.73
Fig. 4(g)	41.99	42.85	42.02	42.74	**42.86**
Fig. 4(h)	33.99	34.90	**35.66**	35.00	35.31
Fig. 4(i)	41.36	42.38	41.80	**41.93**	41.89
Fig. 4(j)	41.65	**42.32**	41.98	41.75	42.05
Fig. 4(k)	38.18	39.32	39.15	39.67	**39.83**
Fig. 4(l)	42.37	43.27	42.69	43.25	**43.56**
Fig. 4(m)	31.98	33.15	35.74	35.22	**37.52**
Fig. 4(n)	36.30	37.63	38.30	37.57	**38.90**
Fig. 4(o)	38.80	39.34	39.87	38.64	**40.02**
Fig. 4(p)	42.27	42.89	43.45	43.33	**43.57**
Fig. 4(q)	40.22	40.92	**41.60**	41.27	41.44
Fig. 4(r)	35.47	36.61	37.04	36.93	**37.19**
Fig. 4(s)	39.17	39.98	40.51	40.63	**40.78**
Fig. 4(t)	39.91	40.68	40.68	40.89	**41.32**
Fig. 4(u)	37.24	38.19	39.35	39.28	**39.45**
Fig. 4(v)	37.55	38.62	38.36	39.08	**39.27**
Fig. 4(w)	42.22	**43.82**	43.08	43.24	43.31
Fig. 4(x)	34.13	34.68	35.56	35.36	**35.63**
Average	37.68	38.43	38.64	38.85	**39.11**

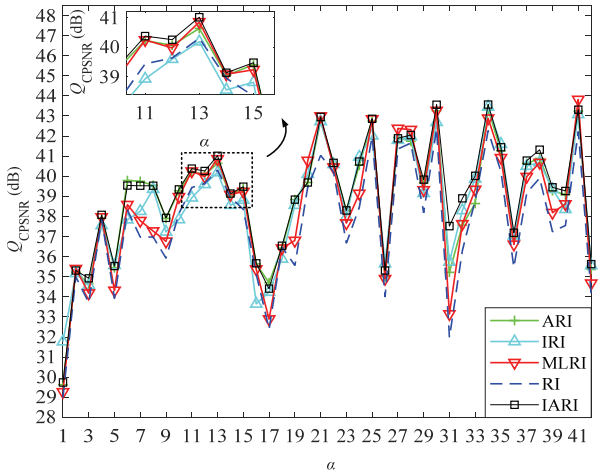

Fig. 8. Comparison of Q_{CPSNR} (dB) among five algorithms.

4.3 Comparison of Subjective Quality of Images

In this section, IMAX03 and IMAX08, as shown in Fig. 3(c) and (h), are selected to illustrate the subjective effects of the five algorithms, as shown in Figs. 9 and 10.

Comparing the flower diagrams of the original figure in Fig. 9(b) as a standard, in the complex marginal area where green grass and pink flowers cross each other, Fig. 9(c)–(e) grass and flowers have a certain degree of mosaic effect. The flower diagrams shown in Fig. 9(f) and (g) have a good recovery effect on the complex edge regions of this position, and the flower color recovery brightness of Fig. 9(g) is closest to Fig. 9(b). Comparing the wave point diagrams in the original figure of Fig. 10(b) as a standard, the recovery effects of the wave point areas of the red tie are not satisfactory in Fig. 10(c)–(e). Compared with the IARI wave point diagram shown in Fig. 10(g), the ARI wave point diagram of Fig. 10(f) basically restores the effect, but the black wave point of some positions produces a significant loss.

In summary, the subjective evaluation proves that the interpolation effect of the IARI algorithm is better than that of other four comparison algorithms.

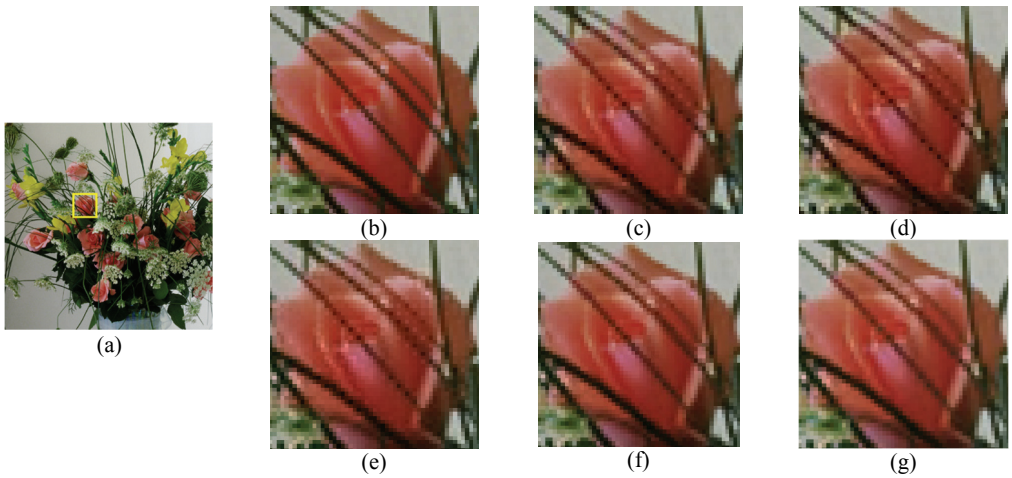

Fig. 9. Experimental results: (a) IMAX03, (b) original color images, (c) demosaicked by RI, (d) demosaicked by MLRI, (e) demosaicked by IRI, (f) demosaicked by ARI, and (g) demosaicked by IARI.

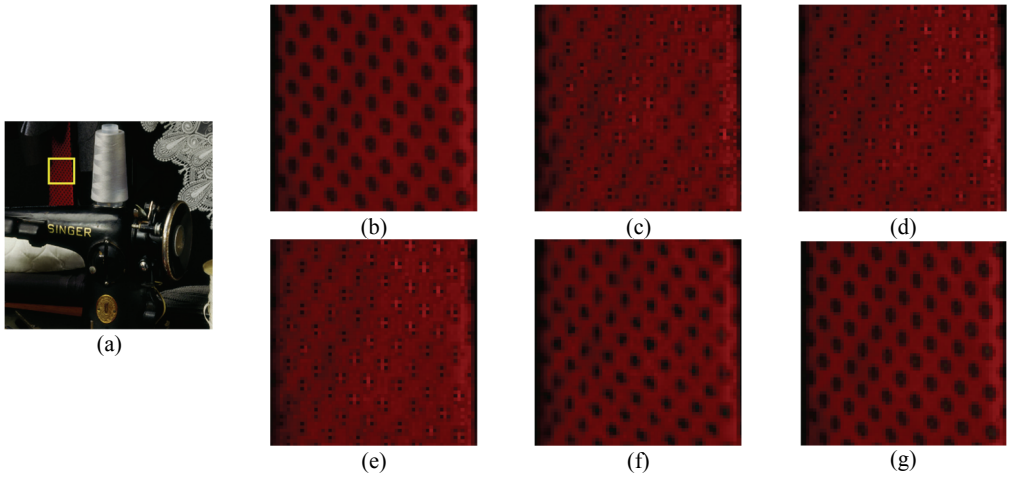

Fig. 10. Experimental results: (a) IMAX08, (b) original color images, (c) demosaicked by RI, (d) demosaicked by MLRI, (e) demosaicked by IRI, (f) demosaicked by ARI, and (g) demosaicked by IARI.

4.4 Comparison of Algorithm Complexity

In the operating environment utilized in this paper, 42 images in IMAX and Kodak datasets are selected, and then each image is performed 15 times by using RI, MLRI, IRI, ARI, and IARI algorithms, respectively, and the average interpolation processing time is shown in Table 3. It can be seen from Table 3 that the average processing time of IARI and ARI is much longer than that of the other three algorithms, and the average time of IARI algorithm is 6.63 seconds longer than that of ARI algorithm. The experimental results show that the IARI algorithm has a higher computational complexity.

Table 3. Average processing time of data set

	RI [7]	MLRI [8]	IRI [9]	ARI [12]	IARI
Average processing time (s)	1.38	2.10	6.44	36.16	42.79

5. Conclusion

This paper focuses on ARI algorithm and illustrates the advantages and disadvantages of the four interpolation algorithms in image demosaicking by analyzing their principles, then proposes an improved algorithm IARI with better interpolation performance. According to the directional difference, IARI algorithm executes ARI algorithm twice on R and B components. By this way, the adaptive interpolation for R and B components is obtained. Experimental results show that the IARI algorithm achieves better performance than other four algorithms both in subjective assessment and objective assessment, especially in the complex edge area and color brightness recovery. The innovation of this paper is that the proposed IARI algorithm divides R and B components interpolation process into two categories according to the directional difference, then performs ARI algorithm interpolation twice according to different categories. Compared with the method by using ARI algorithm directly for both types of interpolation directions, the IARI algorithm is easier to operate. The improved algorithm proposed in this paper solves the problem of incomplete adaptive processing of R and B components in ARI algorithm

interpolation process effectively. However, there are still some disadvantages in IARI algorithm. For example, the higher complexity of the IARI algorithm greatly sacrifices the interpolation speed of the algorithm, and the calculation time lasts a little long. Therefore, in the future, more research for improving the performance of IARI algorithm will be done. Under the premise of maintaining high performance, the complexity of IARI algorithm will be reduced as much as possible, therefore, the processing time of IARI algorithm will be reduced. In Eq. (11), the same parameters $(m,n)=(2,1)$ as the calculated G component are selected, which might make the calculated values of R and B components not optimal and not accurate. Therefore, the setting of optimal parameters in R and B components processes should also be studied. The detailed future works are as follows. In [17], two important issues in the iterative RI process need to be further investigated: (1) the window size of the regression filter, and (2) the stopping criterion for the RI iterative process. Based on the reconstructed G component, the optimal parameters of R and B components should be determined using the description in [7], and then the optimal iteration coefficient will be determined.

References

[1] 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, no. 1, pp. 44-54, 2005.

[2] B. E. Bayer, "Color imaging array," U.S. Patent 3971065, Jul. 20, 1976.

[3] P. Longere, X. Zhang, P. B. Delahunt, and D. H. Brainard, "Perceptual assessment of demosaicing algorithm performance," *Proceedings of the IEEE*, vol. 90, no. 1, pp. 123-132, 2002.

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

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

[6] I. Pekkucuksen and Y. Altunbasak, "Gradient based threshold free color filter array interpolation," in *Proceedings of the IEEE International Conference on Image Processing*, Hong Kong, China, 2010, pp. 137-140.

[7] D. Kiku, Y. Monno, M. Tanaka, and M. Okutomi, "Residual interpolation for color image demosaicking," in *Proceedings of the IEEE International Conference on Image Processing*, Melbourne, Australia, 2013, pp. 2304-2308.

[8] D. Kiku, Y. Monno, M. Tanaka, and M. Okutomi, "Minimized-Laplacian residual interpolation for color image demosaicking," in *Proceedings of SPIE 9023: Digital Photography X*. Bellingham, WA: International Society for Optics and Phtonics, 2014.

[9] W. Ye and K. K. Ma, "Color image demosaicing using iterative residual interpolation," *IEEE Transactions on Image Processing*, vol. 24, no. 12, pp. 5879-5891, 2015.

[10] Y. Kim and J. Jeong, "Four-direction residual interpolation for demosaicking," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 26, no. 5, pp. 881-890, 2016.

[11] K. Yu, C. Wang, S. Yang, Z. Lu, and D. Zhao, "An effective directional residual interpolation algorithm for color image demosaicking," *Applied Sciences*, vol. 8, no. 5, article no. 680, 2018.

[12] Y. Monno, D. Kiku, M. Tanaka, and M. Okutomi, "Adaptive residual interpolation for color and multispectral image demosaicking," *Sensors*, vol. 17, no. 12, article no. 2787, 2017.

[13] Z. Ni, K. K. Ma, H. Zeng, and B. Zhong, "Color image demosaicing using progressive collaborative representation," *IEEE Transactions on Image Processing*, vol. 29, pp. 4952-4964, 2020.

[14] B. Sun, N. Yuan, and Z. Zhao, "A hybrid demosaicking algorithm for area scan industrial camera based on fuzzy edge strength and residual interpolation," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 6, pp. 4038-4048, 2020.

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

[16] J. F. Hamilton and J. E. Adams, "Adaptive color plane interpolation in single color electronic camera," U.S. Patent. 5629734, May 13, 1997.

[17] W. Ye and K. K. Ma, "Image demosaicing by using iterative residual interpolation," in *Proceedings of the IEEE International Conference on Image Processing*, Paris, France, 2014, pp. 1862-1866.

[18] X. Li, B. Gunturk, and L. Zhang, "Image demosaicing: A systematic survey," in *Proceedings of SPIE 6822: Visual Communications and Image Processing*. Bellingham, WA: International Society for Optics and Photonics, 2008.

Chenbo Liu **https://orcid.org/0000-0002-0095-5769**

She was born in Shandong province, China, in 1998. She was admitted into the College of Engineering, Qufu Normal University (Rizhao campus), in 2017. Now she is a fourth year undergraduate student and her major is automation. Her current research interests include digital image processing, deep learning, and artificial intelligence.