Color Image Demosaicing Using Progressive Collaborative Representation

Zhangkai Ni, *Student Member, IEEE*, Kai-Kuang Ma, *Fellow, IEEE*, Huanqiang Zeng, *Senior Member, IEEE*, and Baojiang Zhong, *Senior Member, IEEE*

***Abstract*— In this paper, a *progressive collaborative representation* (PCR) framework is proposed that is able to incorporate any existing color image demosaicing method for further boosting its demosaicing performance. Our PCR consists of two phases: (i) *offline* training and (ii) *online* refinement. In phase (i), multiple *training-and-refining* stages will be performed. In each stage, a new dictionary will be established through the learning of a large number of *feature-patch* pairs, extracted from the demosaicked images of the current stage and their corresponding original full-color images. After training, a *projection matrix* will be generated and exploited to refine the current demosaicked image. The updated image with improved image quality will be used as the input for the next training-and-refining stage and performed the same processing likewise. At the end of phase (i), all the projection matrices generated as above-mentioned will be exploited in phase (ii) to conduct online demosaicked image refinement of the test image. Extensive simulations conducted on two commonly-used test datasets (i.e., IMAX and Kodak) for evaluating the demosaicing algorithms have clearly demonstrated that our proposed PCR framework is able to constantly boost the performance of any image demosaicing method we experimented, in terms of objective and subjective performance evaluations.**

***Index Terms*— Image demosaicing, color filter array (CFA), residual interpolation, progressive collaborative representation.**

I. Introduction

COLOR image demosaicing has been an important issue in the field of image processing. It plays a crucial role in producing high-quality color imagery from a single-sensor digital camera. The demosaicing processing aims to reconstruct a full-color image from the acquired *mosaicked* image by estimating the values of the other two missing color components at each pixel position. In general, a full-color image is composed of three primary color components (i.e., red, green, and blue, which are denoted by R, G, and B, respectively) at each pixel location. However, considering the cost and physical size, almost all consumer-grade digital cameras exploit a single image sensor covered with a *color filter array* (CFA) on the sensor's surface such that only *one* color-component value can be registered at each pixel location on the CFA. The most widely-used CFA pattern is the so-called Bayer's pattern [1]. Such recorded CFA data is commonly termed as a *mosaicked* image. In order to produce a *full* color image from the mosaicked image, the other two missing color component values at each pixel's location are required to be estimated, and this process is called the *demosaicing*.

Due to strong correlations existing among three color channels in nature, many demosaicing algorithms estimate the missing color-component values based on the *color difference* (CD) fields (e.g., R-G or B-G) [3]–[9]. The effectiveness of using this strategy is due to the fact that each generated CD field tends to yield a fairly smooth data field that is highly beneficial to the estimation of those missing color-component values. Since the Bayer's pattern [1] has twice the number of the available G channel samples as that of the R channel and of the B channel respectively, the reconstruction of a *full* G channel is thus considered as the most crucial and first step to achieve, from which the full R channel and the full B channel are then constructed, respectively.

Hamilton *et al.* [3] proposed to estimate the absent green-channel pixel values along the horizontal and vertical directions individually, based on the first-order derivatives of the sub-sampled G channel and the second-order derivatives of the sub-sampled R and B channels. Zhang *et al.* [4] proposed *directional linear minimum mean square error estimation* (DLMMSE) that optimally combines color differences along the horizontal and vertical directions. Pekkucuksen *et al.* proposed the *gradient-based threshold-free* (GBTF) [5] and their improved algorithm, *multiscale gradients-based* (MSG) [8], which is able to yield more pleasant visual results. There are many other CD-based demosaicing methods [2], [6], [10]–[21]. For a survey of the CD-based demosaicing algorithms, refer to [22].

The more promising and effective demosaicing approach is to conduct the demosaicing on the *prediction residual* (PR) fields, rather than on the CD fields. Based on the PR field, this new category of demosaicing algorithms is

Manuscript received July 27, 2019; revised January 8, 2020; accepted February 6, 2020. Date of publication March 4, 2020; date of current version March 12, 2020. This work supported by Ministry of Education, Singapore, under Grant AcRF TIER 1 2017-T1-002-110 and Grant TIER 2 2015-T2-2-114. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Hitoshi Kiya. *(Corresponding author: Kai-Kuang Ma.)*

Zhangkai Ni was with the School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore 639798. He is now with the Department of Computer Science, City University of Hong Kong, Hong Kong (e-mail: eezkni@gmail.com).

Kai-Kuang Ma is with the School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore 639798 (e-mail: ekkma@ntu.edu.sg).

Huanqiang Zeng is with the School of Information Science and Engineering, Huaqiao University, Xiamen 361021, China (e-mail: zeng0043@hqu.edu.cn).

Baojiang Zhong is with the School of Computer Science and Technology, Soochow University, Suzhou 215006, China (e-mail: bjzhong@suda.edu.cn).

Digital Object Identifier 10.1109/TIP.2020.2975978

called the *residual interpolation* (RI) methods [2], [18]–[21], in which the *guided filter* (GF) [23] is exploited to estimate the missing color components. That is, the estimation of a certain channel is generated under the *guidance* of another channel. The *residuals* (i.e., estimation errors), resulted by the GF, are the difference yielded between the sensor-registered pixel values (i.e., the ground truth) and the pixel values estimated by the GF. Compared with the CD-based approach, the RI-based methods are advantageous on both *peak signal-to-noise* (PSNR) and subjective visual quality. Since a smoother image field is more easy to conduct interpolation, the success of the RI-based approach lies in the fact that its resulted PR data field is smoother than the CD data field. Furthermore, among all existing RI-based methods, the IRI [2], [20] delivers attractive demosaicing performance while maintaining reasonable computational complexity.

In recent years, many image processing algorithms have been developed by leveraging the potentials of *convolutional neural network* (CNN) to alleviate the dependence on the hand-crafted priors. Tan *et al.* [24] proposed a two-stage CNN-based demosaicking algorithm, while Cui *et al.* [25] proposed a three-stage CNN-based demosaicking method. Tan *et al.* [26] proposed a multiple CNN structure for demosaicking. Besides, Gharbi *et al.* [27] and Kokkinos *et al.* [28] designed a fully CNN to perform joint demosaicking and denoising.

In this paper, a general image refinement framework, called the *progressive collaborative representation* (PCR), is proposed. The block diagram of the PCR is depicted in Fig. 1, where the IRI is used for demonstration while any other demosaicing algorithm can be used in this framework for improving its demosaicked image quality. Therefore, the proposed PCR is a general framework for conducting image quality refinement, which consists of two phases: (i) offline training and (ii) online refinement. The learned projection matrices computed in the phase (i) will be exploited to progressively refine the image quality of the test image in the phase (ii). Our PCR is developed based on the intuition that the loss or distortion of image details in a demosaicked image can be recovered through a sequence of training-and-refining stages to correct these errors, due to algorithm's limitations on handling adverse conditions such as spectral correlation among three color channels, low-lighting incurred noise, and sophisticated image contents. Given a stage, it can be viewed as a *corrector* of the previous stage, exercising the similar strategy of the *prediction-and-correction* methodology [29]. Consequently, any large prediction errors that have not been effectively corrected in the current stage would have more chances to be further corrected in the subsequent stages, since each stage will re-compute the remaining prediction errors and re-learn from them for conducting further corrections via the generated projection matrix.

The remainder of this paper is organized as follows. In Section II, the proposed PCR demosaicing method is presented in detail. In Section III, extensive performance evaluations of the proposed PCR and other state-of-the-arts are performed and compared. Section IV concludes the paper.

II. Progressive Collaborative Representation (PCR) Image Enhancement Framework

A. Overview

In this paper, a *progressive collaborative representation* (PCR) framework is proposed and exploited for enhancing demosaicked image quality, as depicted in Fig. 1. It consists of two phases: (i) offline training and (ii) online demosaicing. In the offline phase (i), multiple *training-and-refining* stages will be performed. The goal is to generate a *projection matrix* at the end of each training stage such that it can be exploited for refining the demosaicked images obtained from the previous stage. The refined demosaicked images will be used for the next training-and-refining stage.

In the online phase (ii), the given mosaicked image is subject to be demosaicked. For that, a chosen demosaicing algorithm (i.e., the IRI [2] as shown in Fig. 1) will be applied to produce an *initial* demosaicked image as the starting stage for conducting progressive refinements—based on the projection matrices supplied from the offline training stage. Our proposed PCR framework is detailed in the following subsections, respectively.

B. Phase 1: Compute the Projection Matrices via Offline Training

As shown in Fig. 1, the original full-color images (i.e., the ground truth) O_i (where $i = 1, 2, \ldots, N_O$), the simulated mosaicked images M_i, and their initially-generated *demosaicked* images $D_i^{(1)}$ are prepared for the training stage, from which the original O_i and the demosaicked images $D_i^{(1)}$ are used to form *collaborative representations* as the inputs for conducting training for generating the first projection matrix $P^{(1)}$. Note that the superscript (1) here denotes the first iteration stage. Such practice will be applied to all other defined variables that also involve iteration index likewise. That is, symbol (n) on the superscript of a variable denotes the mentioned quantity obtained in the n-th iteration.

The generated $P^{(1)}$ is then used to refine the demosaicked images $D_i^{(1)}$ of the current (first) stage; thus, images $D_i^{(1)}$ are further refined and denoted as $D_i^{(2)}$, which is the input of the subsequent stage. Such *training-and-refining* process will be iteratively performed for N times in the training stage. At the end of phase (i), a set of *projection matrices* $P^{(n)}$ (where $n = 1, 2, \ldots, N$) will be generated and to be used in phase (ii) for performing online demosaicing of the given mosaicked image. For the above-mentioned, some important notes are highlighted as follows.

First, the simulation of the mosaicked images M_i are generated from the ground-truth images O_i by subsampling each image according to the Bayer's pattern [1]. Second, due to its superior demosaicing performance, the *iterative residual interpolation* (IRI) [2] is adopted in our work for generating the *initial* demosaicked images $D_i^{(1)}$, respectively. (The same IRI will be used in the online demosaicing stage as well.) The proposed PCR is *generic* in the sense that one might

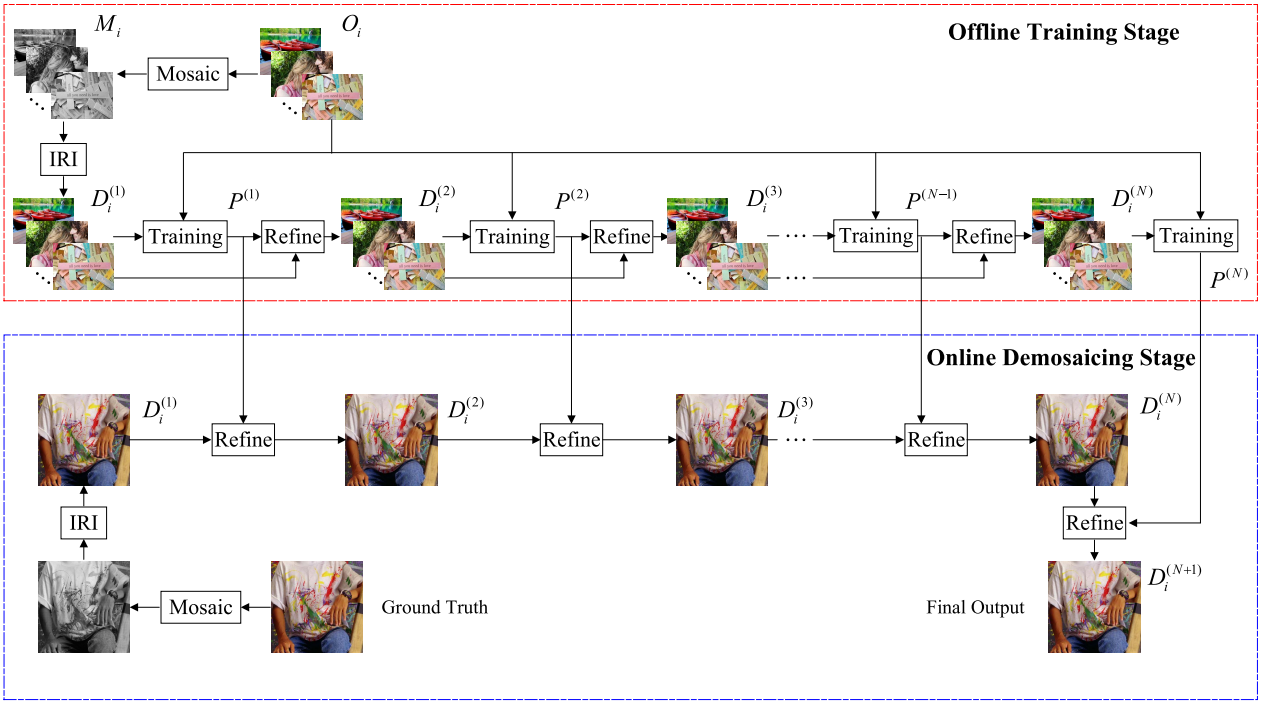


Fig. 1. The proposed *progressive collaborative representation* (PCR) framework for progressively refining the demosaicked color image through multiple stages of training-and-refining (or prediction-and-correction) strategy. It consists of two phases; offline training (top part) and online refinement (bottom part). At each stage in the offline training, a *projection matrix* will be generated to refine the demosaicked images of the current stage, which will be used as the input of the next refinement stage. This projection matrix will be also used in the corresponding stage in the online refinement phase. The state-of-the-art *iterative residual interpolation* (IRI) [2] depicted in this figure is used for demonstration, while any other demosaicing algorithm can be used here for improving its demosicked image's quality.

exploit any other demosaicing algorithm to replace the IRI in both offline training and online testing stages to boost the performance of the considered demosaicing algorithm. Third, two pre-processing steps—(i) feature extraction and (ii) dimensionality reduction (not shown in Fig. 1 for the simplicity of drawing) had been applied to the feature patches generated from the demosaicked images $D_i^{(1)}$ for learning the dictionary. This is to be further detailed using the first iteration for illustration as follows.

The IRI-generated demosaicked images $D_i^{(1)}$ are filtered by using the 1D Roberts and Laplacian high-pass filters performed in the horizontal and vertical directions to extract features, followed by segementing the filtered images into 3×3 image patches. These patches are collectively denoted as the *feature patches* of the demosaicked images $D_i^{(1)}$, respectively. Further note that the above-described steps for generating feature patches will be applied to three color channels (i.e., R, G, and B) separately, followed by concatenating their resulted feature patches together. With consideration of the computational complexity, the *principal component analysis* (PCA) algorithm is further applied to these feature patches for reducing their dimensionality while preserving 99.9% of their average energy. After the PCA process, these feature patches $y_i^{(1)}$ (where $i = 1, 2, \ldots, N_y$) for the respective demosaicked images $D_i^{(1)}$ are used to learn the dictionary $\Xi^{(1)}$ by following a similar processing pipeline as performed in the K-SVD method [30]; i.e., the dictionary $\Xi^{(1)}$ and c_i are required to be generated simultaneously via the following:

$$\left[\Xi^{(1)}, \{c_i\}\right] = \underset{[\Xi^{(1)},\{c_i\}]}{\arg\min} \sum_i \left\| y_i^{(1)} - \Xi^{(1)} \cdot c_i \right\|^2, \quad \text{s.t. } \|c_i\|_0 \le L \text{ and } i = 1, 2, \ldots, N_y, \quad (1)$$

where c_i are the coefficients corresponding to the feature patch $y_i^{(1)}$ and $L = 3$ is the maximal sparsity set for establishing the dictionary $\Xi^{(1)}$.

In our work, the color image demosaicing refinement is considered as a *collaborative representation* problem [31]. That is, given an IRI-demosaicked image, the feature patches $x_i^{(1)}$ are obtained from the image by processing it in a similar way as that of obtaining $y_i^{(1)}$ described previously, and this problem can be boiled down to a least-squares regression problem and regularized by the l_2-norm of the coefficient vector as follows.

First, we need to establish a large number of feature-patch pairs $\{f_{D_i}^{(1)}, f_{E_i}^{(1)}\}$ (where $i = 1, 2, \ldots, N_f$), where $f_{D_i}^{(1)}$ are extracted from a scaled pyramid representation of the demosaicked images $D_i^{(1)}$, while $f_{E_i}^{(1)}$ are generated from the *error* images $E_i^{(1)}$, which is computed by subtracting the demosaicked image from the original full-color image. Further note that no dimensionality reduction will be applied to $f_{E_i}^{(1)}$. Next, we need to search the *neighborhood* for each atom $d_k^{(1)}$ of the dictionary $\Xi^{(1)}$ as follows.

For each atom, the nearest neighborhoods $\{N_{D_k}^{(1)}, N_{E_k}^{(1)}\}$ of the demosaicked images and the corresponding full-color images will be identified from the $\{f_{D_i}^{(1)}, f_{E_i}^{(1)}\}$. Here, the absolute value of the dot product between the atom $d_k^{(1)}$ and each individual feature patch in $f_{D_i}^{(1)}$ will be computed to measure the degree of similarity; that is,

$$\delta\left(d_k^{(1)}, f_{D_i}^{(1)}\right) = \left|\left[d_k^{(1)}\right]^T \cdot f_{D_i}^{(1)}\right|. \tag{2}$$

Thus, the above-mentioned collaborative representation problem can be formulated as

$$\min_{\omega} \left\| x_i^{(1)} - N_{D_k}^{(1)} \cdot \omega \right\|_2^2 + \lambda \left\| \omega \right\|_2, \tag{3}$$

where $N_{D_k}^{(1)}$ is the identified neighborhood from the demosaicked image, ω is a regularization coefficient of $x_i^{(1)}$ over $N_{D_k}^{(1)}$, and λ is used to alleviate the ill-posed problem. The above-stated equation has a closed-form solution, which is given by [32]

$$\omega = \left(\left[N_{D_k}^{(1)}\right]^T \cdot N_{D_k}^{(1)} + \lambda \cdot \mathbf{I}\right)^{-1} \cdot \left[N_{D_k}^{(1)}\right]^T \cdot x_i^{(1)}. \tag{4}$$

where $\mathbf{I}$ is the identity matrix.

The demosaicked image patches can be refined by applying the same coefficients ω to the corresponding full-color image feature patch neighborhoods $N_{E_k}^{(1)}$. This is based on the assumption that the feature patches $x_i^{(1)}$ and the corresponding refined demosaicked image feature patch share the same coefficients ω over $N_{D_k}^{(1)}$ and $N_{E_k}^{(1)}$, respectively [32]. Thus, the demosaicked image feature patch can be refined by computing the following formula:

$$x_i^{(2)} = N_{E_k}^{(1)} \cdot \left(\left[N_{D_k}^{(1)}\right]^T \cdot N_{D_k}^{(1)} + \lambda \cdot \mathbf{I}\right)^{-1} \cdot \left[N_{D_k}^{(1)}\right]^T \cdot x_i^{(1)}. \tag{5}$$

This equation can be formulated as $x_i^{(2)} = p_k^{(1)} \cdot x_i^{(1)}$; i.e.,

$$p_k^{(1)} = N_{E_k}^{(1)} \cdot \left(\left[N_{D_k}^{(1)}\right]^T \cdot N_{D_k}^{(1)} + \lambda \cdot \mathbf{I}\right)^{-1} \cdot \left[N_{D_k}^{(1)}\right]^T. \tag{6}$$

The projection matrix $P^{(1)}$ is composed of matrices $p_k^{(1)}$ for $k = 1, 2, \ldots, N_d$ and used to update $D_i^{(1)}$ (as the input) for yielding a *refined* $D_i^{(2)}$ (as the output). The above-detailed process will be repeated for N times. At the end of the training stage, N projection matrices will be generated. The entire learning process of phase (i) is summarized in Algorithm 1.

C. Phase 2: Progressive Refinements of Demosaicked Images

Refer to bottom part of Fig. 1, the objective of phase (ii) is to progressively improve the image quality of the initially demosaicked image through multiple stages of refinements with the use of the projection matrices. For discussion, denote the original (i.e., the ground truth), its mosaicked images, and the demosaicked images as I_i, M_i, and D_i, respectively, where the M_i is subsampled from the image I_i according to the Bayer's pattern [1]. Based on the M_i, the demosaicing algorithm IRI [2] is exploited for generating a high-quality demosaicked image $D_i^{(1)}$, which is subject to be further refined for improving its image quality. Note that the $D_i^{(1)}$ here is the IRI result of the test images rather than training image. In each of the subsequent refinement stages, the corresponding projection matrix $\{P^{(n)}\}$ with the same iteration index n and supplied from the offline training stage will be exploited to improve the image quality of the refined image from the previous stage.

In the n-th iteration, the computed feature patches $f_{D_i}^{(n)}$ (for $i = 1, 2, \ldots, N_I$) and the corresponding demosaicked image patches $z_i^{(n)}$ from the $D_i^{(n)}$ will be used to reconstruct $D_i^{(n+1)}$. For each feature patch $f_{D_i}^{(n)}$, its nearest atom $d_k^{(n)}$ in $\Xi^{(n)}$ via (2) will be identified. Then, the refined feature patch $f_{D_i}^{(n+1)}$ is computed by

$$f_{D_i}^{(n+1)} = p_k^{(n)} \cdot f_{D_i}^{(n)}. \tag{7}$$

The *refined* demosaicked image patch $z_i^{(n+1)}$ is then obtained by adding the refined feature patch $f_{D_i}^{(n+1)}$ in (7) to the

Algorithm 1 Compute the Projection Matrices via Offline Training

Input:
A set of full-color training images O_i (where $i = 1, 2, \ldots, N_O$).

Output:
The projection matrices $P^{(n)}$ (where $n = 1, 2, \ldots, N$).

Steps:

1: Subsampling the original full-color training images O_i to generate their *mosaicked* versions M_i according to the Bayer's pattern [1], followed by demosaicing each mosaicked image M_i using the IRI [2] to obtain $D_i^{(1)}$.
2: **for** n = 1; $n \leq N$; $n = n + 1$; **do**
3: Extract *feature patches* from $D_i^{(n)}$ by extracting their Robert's and Laplacian high-pass image features. Reduce the dimensionality of these feature patches by applying the PCA algorithm to obtain $y_i^{(n)}$ (where $i = 1, 2, \ldots, N_y$).
4: Train the dictionary $\Xi^{(n)}$ based on $y_i^{(n)}$ using the K-SVD method via (1);
5: Establish a large number of feature-patch pairs $\{f_{D_i}^{(n)}, f_{E_i}^{(n)}\}$ (where $i = 1, 2, \ldots, N_f$) from the training image pairs $\{D_i^{(n)}, O_i\}$ by extracting the Roberts and Laplacian high-pass image features of $D_i^{(n)}$ and subtracting the demosaicked image from the corresponding O_i, respectively.
6: **for** $k = 1$; $k \leq N_d$; $k = k + 1$; **do**
7: Search the demosaicked image neighborhoods $N_{D_k}^{(n)}$ from $f_{D_i}^{(n)}$ for $d_k^{(n)}$.
8: Search the corresponding $N_{E_k}^{(n)}$ from $f_{E_i}^{(n)}$.
9: **end for**
10: Compute the n-th projection matrix $P^{(n)}$ via (6).
11: Update $D_i^{(n)}$ to $D_i^{(n+1)}$ via the projection matrix $P^{(n)}$.
12: **end for**
13: Output the projection matrices $P^{(n)}$.

Algorithm 2 Conduct Online Demosaicing With Progressive Refinements

Input:
A test image I_i and a set of projection matrices $\{P^{(n)}\}$, for $n = 1, 2, \ldots, N$.
Output:
The demosaicked image of the test input image, $D_i^{(N+1)}$.
Steps:
1: Subsample I_i to M_i according to the Bayer's pattern to yield mosaicked image, followed by demosaicing M_i to $D_i^{(1)}$ using the IRI [2] algorithm.
2: **for** $n = 1$; $n \leq N$; $n = n + 1$; **do**
3: Generate demosaicked image patches $z_i^{(n)}$ (for $i = 1, 2, \ldots, N_I$) and extract high-pass feature patches $f_{D_i}^{(n)}$ from $z_i^{(n)}$.
4: Reduce the dimensionality of $f_{D_i}^{(n)}$ using the PCA algorithm.
5: **for** $i = 1$; $i \leq N_I$; $i = i + 1$; **do**
6: Search $f_{D_i}^{(n)}$ with respect to $\Xi^{(n)}$ to identify the nearest-neighbor atoms via (2).
7: Compute the refined feature patches $f_{D_i}^{(n+1)}$ via (7).
8: Reconstruct the refined image patch $z_{D_i}^{(n+1)}$ via (8).
9: **end for**
10: Combine all the refined demosaicked image patches $z_i^{(n+1)}$ and average the pixel values on those overlapping areas to form the refined demosaicked image $D_i^{(n+1)}$.
11: **end for**
12: Output the final demosaicked image $D_i^{(N+1)}$.

demosaicked image patch $z_i^{(n)}$; that is,

$$z_{D_i}^{(n+1)} = f_{D_i}^{(n+1)} + z_i^{(n)}. \tag{8}$$

By combing all the refined patches $z_i^{(n+1)}$ followed by averaging the intensity values on the overlapping areas, the final refined demosaicked image images $D_i^{(n+1)}$ will be generated. The entire phase (ii) is summarized in Algorithm 2.

Lastly, the above-described progressive refinement processing will be conducted for each of three color channels separately.

III. Experiments

A. Experiment Settings

1) Training Dataset: For the learning-based demosaicing methods, the training dataset exploited for training has a direct impact on the quality of the demosaicked images. In our experiments, the proposed PCR is trained by using the same 100 training images as the ones used in [17], and these training images do not adopt any data augmentation (i.e., rotation and flipping) for increasing the size of the training dataset.

2) Testing Dataset: The IMAX dataset (18 images) and Kodak dataset (24 images) are used in our experiments, since they have been widely adopted for assessing the performance of demosaicing methods (e.g., [2], [17], [21]). Note that the images in the IMAX dataset have weaker spectral correlations among three color channels and are considered to be more challenging, while the images in the Kodak dataset have stronger correlations among three color channels [33]. Each full-color test image is firstly sub-sampled according to the Bayer's pattern [1], followed by conducting the demosaicing process.

3) Evaluation Criteria: For objectively evaluating the performance of the demosaicing methods under comparisons, two evaluation metrics are used; i.e., the *color peak signal-to-noise ratio* (CPSNR) and the *structural similarity* (SSIM) [34]. The former is applied to compute the intensity differences between the individual channel of the demosaiced images and the original images (i.e., the ground truth); that is,

$$\text{CPSNR} = 10 \times \log_{10}\left(\frac{255^2}{\text{CMSE}}\right), \text{where}$$
$$\text{CMSE} = \frac{\sum_{C\in\{R,G,B\}}\sum_i^H\sum_j^W \parallel I_o^C(i,j) - I_d^C(i,j) \parallel_2^2}{3 \times H \times W}, \tag{9}$$

symbol $||\cdot||_2$ denotes the l_2 norm of a vector, parameters H and W denote the height and the width of the image, respectively. Note that both $I_o^C(i, j)$ and $I_d^C(i, j)$ are vectors, representing the R, G, or B values at the pixel (i, j) from the original image and the demosaiced image, respectively. To further evaluate the image quality from the perceptual viewpoint of the human visual system, the SSIM is computed as another supporting performance evaluation measurement to reflect the similarity yielded between the original image and the demosaicked images. The SSIM considers the degradations incurred on the demosaicked image's structure, instead of the pixel-intensity differences only. Three types of similarities are concerned in the SSIM—that is, the luminance similarity, the contrast similarity, and the structural similarity. For a joint measurement of the SSIM, the *average* SSIM is obtained by averaging the SSIM values individually obtained from the R, G, and B channels. Note that the higher the SSIM value, the better the perceptual quality of the demosaicked image.

4) Default Settings for the PCR's Parameters: Unless otherwise specified, the simulations conducted for the performance evaluation of our proposed PCR adopted the following default settings. The size of the image patches is set to 3×3 pixels with an overlap of 2 pixels between adjacent patches. Four 1D high-pass filters (i.e., $f_1 = [-1, 0, 1]$, $f_2 = [-1, 0, 1]^T$, $f_3 = [1, 0, -2, 0, 1]$, and $f_4 = [1, 0, -2, 0, 1]^T$) are used to extract the high-frequency details for each image patch, followed by applying the PCA for reducing the dimensionality. The same dictionary training method as described in [32] and [30] is exploited for the PCR; i.e., 4,096 atoms for each dictionary, a neighborhood size of 2,048 training samples, and 5 millions of the training samples from the demosaicked image and original image patches. The algorithm's performance resulted by various experimental settings will be investigated in the following sub-sections.

B. Performance Analysis

To evaluate the performance, the proposed PCR is compared with ten state-of-the-art demosaicing methods: the *learned*

TABLE I

AVERAGE PSNR AND CPSNR RESULTS (IN dB) OBTAINED FROM THE KODAK AND THE IMAX DATASETS. THE FIRST-RANKED, THE SECOND-RANKED, AND THIRD-RANKED PERFORMANCE IN EACH COLUMN IS HIGHLIGHTED IN BOLD WITH RED, BLUE, AND BLACK COLORS, RESPECTIVELY

Methods	IMAX				Kodak				IMAX+Kodak			
	PSNR			CPSNR	PSNR			CPSNR	PSNR			CPSNR
	R	G	B		R	G	B		R	G	B	
LSSC [14]	36.0412	38.8235	34.7159	36.1698	**40.5321**	**44.3148**	**40.6485**	**41.4416**	**38.6025**	**41.9568**	**38.1142**	**39.1769**
GBTF [5]	33.5478	36.5706	32.7165	33.9280	39.4697	43.1112	39.8389	40.4225	36.9318	40.3081	36.7864	37.6391
LDI-NAT [6]	36.2754	39.7648	34.3897	36.2023	36.9727	39.4247	37.1075	37.6720	36.6738	39.5704	35.9427	37.0421
MSG [8]	34.3785	37.6532	33.3853	34.7212	40.0630	43.7729	**40.3190**	40.9890	37.6268	41.1502	37.3474	38.3028
RI [18]	36.0722	39.9923	35.3501	36.4752	37.7934	40.9811	37.7856	38.5400	37.0557	40.5573	36.7418	37.6550
MLRI [19]	36.3460	39.8998	35.3648	36.6240	38.8515	41.8112	38.8418	39.5600	37.7777	40.9920	37.3516	38.3017
DDR [17]	37.0884	40.3345	35.6214	37.1475	37.0884	**43.8875**	**40.3548**	**41.0680**	38.2442 38.0488	40.9634 41.3938	37.5364 37.9155	38.5766 38.7421
FR [17]	**37.4748**	**41.0044**	**35.8077**	**37.4745**	**40.1562**	**43.8126**	40.3096	**41.0392**	38.2442 38.2075	40.9634 41.6568	37.5364 **38.0048**	38.5766 38.8828
ARI [21]	**37.4534**	**40.6764**	**36.2137**	**37.5963**	39.2522	42.4076	39.0699	39.9288	**38.4813**	**41.6656**	37.8458	**38.9291**
IRI [2]	36.8048	40.2740	35.7229	37.0597	38.7853	42.3171	39.0882	39.7181	37.9365	41.4415	37.6459	38.5788
IRI-PCR (Ours)	**38.2977**	**41.4566**	**36.4841**	**38.1948**	40.0297	43.5259	39.7813	40.7562	**39.2874**	**42.6391**	**38.3682**	**39.6585**

simultaneous sparse coding (LSSC) [14], the *gradient-based threshold-free* (GBTF) [5], the *local directional interpolation and nonlocal adaptive thresholding* (LDI-NAT) [6], the *multiscale gradients-based* (MSG) [8], the *residual interpolation* (RI) [18], the *minimized-Laplacian residual interpolation* (MLRI) [19], the *directional difference regression* (DDR) [17], the *fused regression* (FR) [17], the *adaptive residual interpolation* (ARI) [21], and the *iterative residual interpolation* (IRI) [2]. Note that the source codes of these methods are all downloaded from their corresponding authors.

1) Objective Performance Analysis: Firstly, Table I compares the proposed method with the state-of-the-art algorithms in terms of the average PSNR and CPSNR on the IMAX dataset, the Kodak dataset, and their combined dataset (denoted as "IMAX+Kodak"). Note that the DDR [17] and FR [17] have two different sets of parameters for the IMAX and Kodak datasets, separately. For the "IMAX+Kodak" dataset, we apply each above-mentioned parameter set for the images of this dataset and consequently lead to two sets of experimental results, one for the DDR [17] and the other one for the FR.

From the Table I, one can observe that our proposed PCR (with the use of IRI [2] as the initial demosaicing stage) has achieved the best average CPSNR among all the demosaicing methods under comparison on the IMAX dataset. Moreover, the PCR gains additional 1.1351 dB improvement on the IMAX dataset and 1.0381 dB on the Kodak dataset, when compared to that of IRI [2]. It is clearly to see that the proposed PCR refinement framework effectively improves the performance even for the state-of-the-art method such as the IRI experimented in this case. Considered the second best method, ARI [21], performed on the IMAX dataset, additional 0.5985 dB gain is achieved. Compared with three learning-based methods (i.e., LSSC [14], DDR [17], and FR [17]) experimented on the IMAX dataset, additional performance gain yielded by our proposed PCR are 2.0250 dB, 1.0473 dB, and 0.7203 dB, respectively. Furthermore, Table I also presents the quantitative comparisons on each color channel individually for all methods in terms of the PSNR, from which one can see that the average PSNR obtained from each color channel using our proposed PCR also delivers the best performance on the IMAX dataset. In addition, our proposed approach PCR also achieves the highest average CPSNR on the combined IMAX+Kodak dataset.

Similar to Table I, Table II and Table III compare the proposed method with the same set of state-of-the-art algorithms in terms of the SSIM and MSSIM, respectively. One can still observe that our proposed PCR consistently yields the best performance on the IMAX dataset and the combined IMAX+Kodak dataset. For the Kodak dataset, our proposed method delivers a fairly close performance to the best method LSSC [14].

Lastly, it is worthwhile to point out that several demosaicing methods achieve good performance on the Kodak dataset, but not on the IMAX dataset. For example, the sparse-representation-based LSSC [14] method has delivered the best CPSNR performance on the Kodak dataset. However, its performance drops drastically on the IMAX dataset. The DDR [17] and FR [17] have also shown such trend. This might be due to the well-known fact that the color images from the Kodak dataset have unusually high spectral correlations among three color channels; this has been highlighted in several previous works (e.g., [33]). Since these methods favour to those color images with high degree of spectral correlations, they tend to produce inferior demosaicked results otherwise, such as those images from the IMAX dataset. In contrast, our proposed PCR is much insensitive to spectral correlations among three color channels and thus achieves more robust performance regardless which dataset is used.

2) Subjective Performance Analysis: Besides the superiority on the objective evaluations, our proposed PCR method also shows superiority on the subjective quality assessment. All images from both datasets have been experimented, and the demosaicked images have shown consistent performance trend on both objective and subjective evaluations. Two representative test images from each dataset are selected for conducting visual comparison, since they are more challenging to perform demosaicing. Specifically, Figs. 2-5 show the demosaicked

TABLE II

AVERAGE SSIM RESULTS OBTAINED FROM THE KODAK AND THE IMAX DATASETS. THE FIRST-RANKED, THE SECOND-RANKED, AND THIRD-RANKED PERFORMANCE IN EACH COLUMN IS HIGHLIGHTED IN BOLD WITH RED, BLUE, AND BLACK COLORS, RESPECTIVELY

Methods	IMAX				Kodak				IMAX+Kodak			
	SSIM			Avg.	SSIM			Avg.	SSIM			Avg.
	R	G	B		R	G	B		R	G	B	
LSSC [14]	0.9550	0.9734	0.9274	0.9519	**0.9848**	**0.9925**	0.9835	**0.9869**	0.9720	0.9843	0.9595	0.9719
GBTF [5]	0.9317	0.9558	0.8927	0.9267	0.9827	0.9910	0.9824	0.9853	0.9609	0.9759	0.9439	0.9602
LDI-NAT [6]	0.9601	0.9785	0.9245	0.9543	0.9725	0.9830	0.9714	0.9756	0.9672	0.9811	0.9513	0.9665
MSG [8]	0.9392	0.9629	0.9025	0.9349	0.9843	0.9919	**0.9837**	0.9866	0.9649	0.9795	0.9489	0.9644
RI [18]	0.9592	0.9798	0.9400	0.9597	0.9761	0.9869	0.9734	0.9788	0.9688	0.9838	0.9591	0.9706
MLRI [19]	0.9605	0.9788	0.9389	0.9594	0.9807	0.9889	0.9792	0.9829	0.9721	0.9846	0.9619	0.9728
DDR [17]	0.9636	0.9816	0.9387	0.9613	**0.9849**	**0.9923**	**0.9841**	**0.9871**	0.9738 0.9699	0.9850 0.9816	0.9619 0.9596	0.9736 0.9703
FR [17]	**0.9659**	**0.9832**	0.9407	**0.9633**	**0.9847**	**0.9922**	**0.9839**	**0.9870**	**0.9741** 0.9713	0.9852 0.9836	0.9620 0.9610	**0.9738** 0.9720
ARI [21]	**0.9680**	**0.9829**	**0.9467**	**0.9658**	0.9800	0.9884	0.9772	0.9819	**0.9749**	**0.9860**	**0.9641**	**0.9750**
IRI [2]	0.9633	0.9818	**0.9411**	0.9621	0.9789	0.9891	0.9780	0.9820	0.9722	**0.9861**	**0.9622**	0.9735
IRI-PCR (Ours)	**0.9703**	**0.9843**	**0.9474**	**0.9672**	0.9841	0.9917	0.9819	0.9859	**0.9782**	**0.9884**	**0.9671**	**0.9779**

TABLE III

AVERAGE MSSIM RESULTS ON THE KODAK AND THE IMAX DATASETS. THE FIRST-RANKED, THE SECOND-RANKED, AND THIRD-RANKED PERFORMANCE IN EACH COLUMN IS HIGHLIGHTED IN RED, BLUE, AND BLACK BOLDS, RESPECTIVELY

Methods	IMAX				Kodak				IMAX+Kodak			
	MSSIM			Avg.	MSSIM			Avg.	MSSIM			Avg.
	R	G	B		R	G	B		R	G	B	
LSSC [14]	0.9926	0.9969	0.9869	0.9921	**0.9964**	**0.9990**	**0.9960**	**0.9972**	**0.9948**	**0.9981**	0.9921	0.9950
GBTF [5]	0.9872	0.9942	0.9763	0.9859	0.9952	**0.9987**	0.9952	0.9964	0.9918	0.9968	0.9871	0.9919
LDI-NAT [6]	0.9933	0.9976	0.9868	0.9926	0.9933	0.9977	0.9926	0.9945	0.9933	0.9977	0.9901	0.9937
MSG [8]	0.9892	0.9954	0.9800	0.9882	0.9957	**0.9989**	**0.9956**	**0.9967**	0.9930	0.9974	0.9889	0.9931
RI [18]	0.9934	0.9977	0.9899	0.9937	0.9942	0.9982	0.9933	0.9952	0.9939	0.9980	0.9918	0.9946
MLRI [19]	0.9934	0.9976	0.9895	0.9935	0.9949	0.9985	0.9943	0.9959	0.9942	**0.9981**	0.9923	0.9949
DDR [17]	0.9943	0.9979	0.9897	0.9940	**0.9958**	**0.9989**	**0.9956**	**0.9968**	0.9947 0.9940	**0.9981** 0.9977	0.9923 0.9920	0.9950 0.9946
FR [17]	**0.9946**	**0.9981**	0.9900	0.9942	0.9951	0.9982	0.9953	**0.9967**	0.9947 0.9943	**0.9981** 0.9979	0.9923 0.9923	0.9950 0.9948
ARI [21]	**0.9951**	**0.9980**	**0.9909**	**0.9945**	0.9953	0.9985	0.9946	0.9961	**0.9952**	**0.9983**	**0.9931**	**0.9955**
IRI [2]	0.9945	0.9979	**0.9903**	**0.9943**	0.9950	0.9986	0.9947	0.9961	0.9947	0.9980	**0.9928**	**0.9951**
IRI-PCR (Ours)	**0.9956**	**0.9983**	**0.9912**	**0.9949**	**0.9960**	**0.9989**	**0.9954**	**0.9968**	**0.9958**	**0.9986**	**0.9936**	**0.9960**

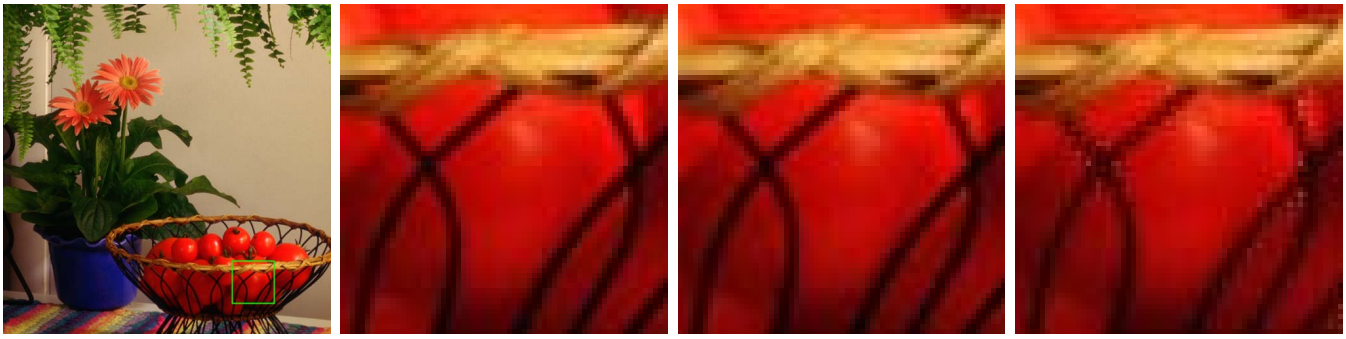
results of four cropped-up sub-images (as indicated by the green-colored frame in each test image) for close-up visual comparisons.

In Fig. 2, one can see that the zoom-in region contains rattan basket (with light brown and dark black colors) and the fruit in red. Due to the fact that the spectral correlations among the three color channels of the color images from the IMAX dataset are much less correlated, the GBTF [5], FR [17], ARI [21] and IRI [2] tend to yield zipper effect. Furthermore, some distinct color artifacts can be observed around the edges of the black rattan in Fig. 2 (d)-(g). Although the sparse-based LSSC [14] method can yield a fairly similar demosaicked image to that of our proposed PCR method, one can easily observe that many details of the PCR-demosaicked image are still superior to and more natural than that of the LSSC [14] with less zipper effect (e.g., the brown rattan area). In fact, our PCR result is nearly identical to the ground truth on this zoom-in image.

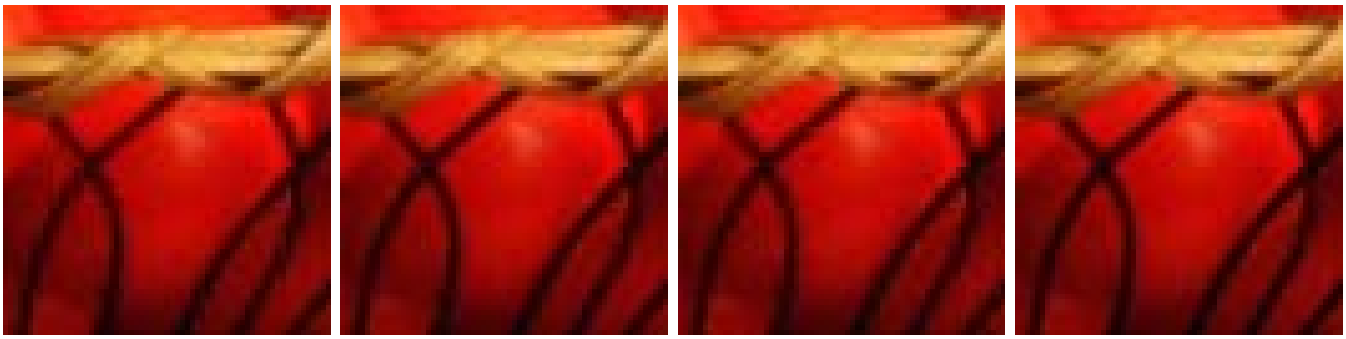
Compared with the interpolation-based methods (i.e., GBTF [5], ARI [21], and IRI [2]), the learning-based demosaicing methods (i.e., LSSC [14], FR [17], and our proposed PCR) have produced much improved demosaicked images. Fig. 3 demonstrates a close-up of the demosaicked image from the IMAX 12. Through comparison, one can see that our proposed PCR delivers the best visual quality on sharp edges and color details. On the contrary, the other methods under comparison produce many visible zipper artifacts and false colors along the edges of drawings, as shown in Fig. 3(c)-(g).

Fig. 4 displays a zoom-in portion of a jalousie window, which is often used to evaluate the demosaicked results of highly-textured regions. One can see that our proposed PCR has achieved distinctly superior demosaicked result, which is almost identical to the original image. On the contrast, one can observe obvious color aliasing and pattern shift resulted by LSSC [14], GBTF [5], FR [17], ARI [21], and IRI [2]. Although the LSSC [14] and GBTF [5] have yielded much better visual results than those in Fig. 4 (e)-(g), however they still produced noticeable false color artifacts, as illustrated in Fig. 4 (c)-(d).

To further evaluate the demosaicked results of those textured areas using our proposed PCR, Figs. 5 demonstrates a zoom-in area that is also highly textured as it has dense edges

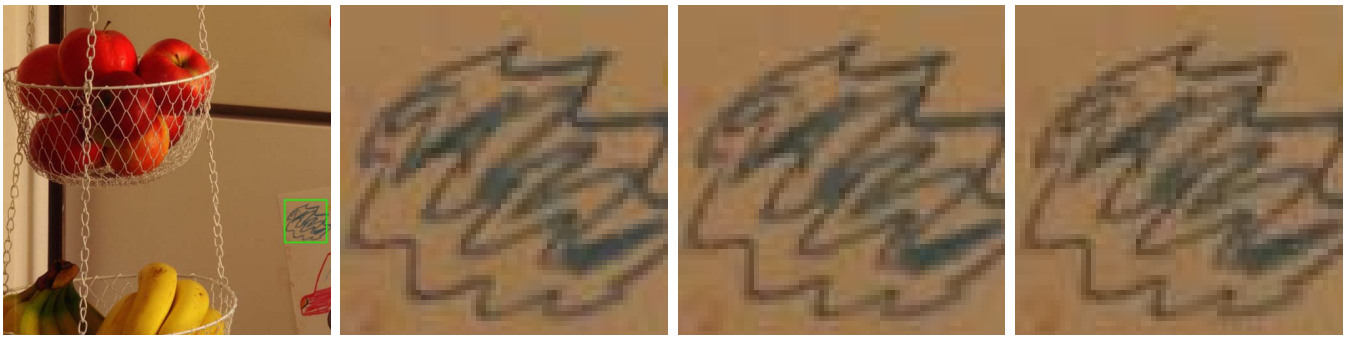


(a) Scaled original IMAX 9 (b) Ground truth (c) LSSC [14] (d) GBTF [5]

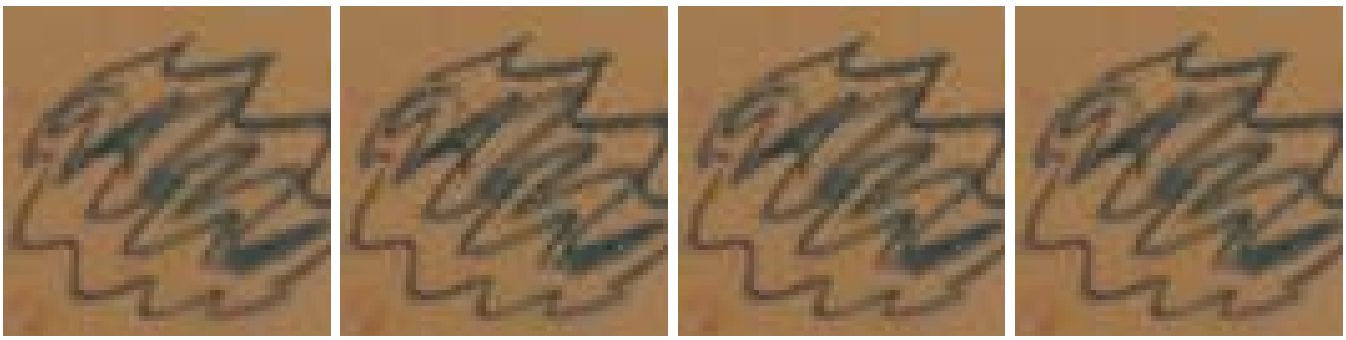


(e) FR [17] (f) ARI [21] (g) IRI [2] (h) IRI-PCR (Ours)

Fig. 2. Visual comparisons for a close-up region on the image "IMAX 9" from the IMAX dataset.



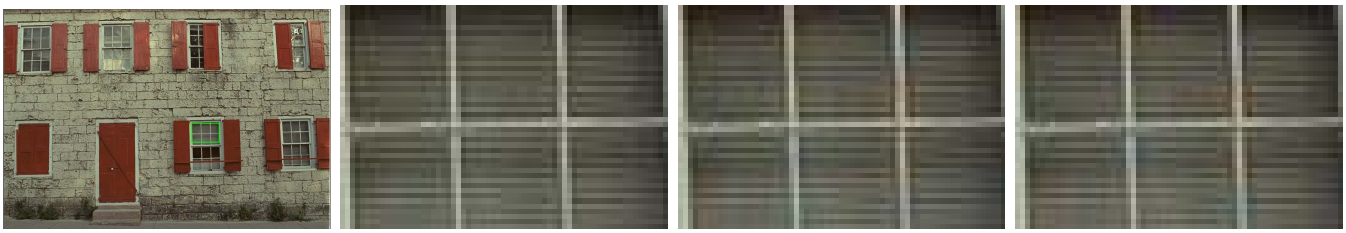
(a) Scaled original IMAX 12 (b) Ground truth (c) LSSC [14] (d) GBTF [5]



(e) FR [17] (f) ARI [21] (g) IRI [2] (h) IRI-PCR (Ours)

Fig. 3. Visual comparisons for a close-up region on the image "IMAX 12" from the IMAX dataset.

clustered in a small region. It is quite clear to see that, all algorithms except our proposed PCR introduce false color artifacts and/or edge distortion. Especially, the LSSC [14], GBTF [5], FR [17] and IRI [2] methods produce the most distinct color artifacts in this case, while the ARI [21] method has delivered much better demosaiced image quality. However, with a closer look, one can see that our proposed PCR has yielded the least amount of artifacts; e.g., refer to the top-right portion of the sub-image in Figs. 5, where the ARI [21] has produced visible color leakage in blue. This study on visual quality has further demonstrated the superiority of our proposed PCR.



(a) Scaled original image (b) Ground truth (c) LSSC [14] (d) GBTF [5]

(e) FR [17] (f) ARI [21] (g) IRI [2] (h) IRI-PCR (Ours)

Fig. 4. Visual comparisons for a close-up region on the image "Kodak 1" from the Kodak dataset.

(a) Scaled original image (b) Ground truth (c) LSSC [14] (d) GBTF [5]

(e) FR [17] (f) ARI [21] (g) IRI [2] (h) IRI-PCR (Ours)

Fig. 5. Visual comparisons for a close-up region on the image "Kodak 8" from the Kodak dataset.

3) Computational Complexity Analysis: To analyze the computational complexity for each demosaicing method, the average running time per image is measured, based on the IMAX and the Kodak datasets. The computer used for conducting our simulation experiments is equipped with an Intel Xeon CPU E5-1630 v4@3.70GHz with 32GBs of RAM, and the software platform we exploited is Matlab R2017b. Note that all the competing demosaicing models are performed under the same test conditions and procedures to have a meaningful and fair comparison. The run-time results of all demosaicing methods are documented in Table IV. For a more intuitive comparison between the computational complexity and demosaicing performance, Fig. 6 shows the CPSNR versus the running time of all demosaicing methods for comparison.

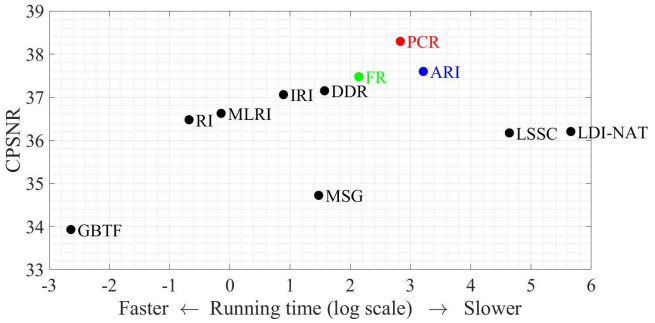


Fig. 6. CPSNR versus running time (in log scale) of different methods performed on the IMAX dataset.

From Table IV, it can be seen that the proposed PCR is computationally more expensive than most non-learning-based methods (i.e., GBTF [5], MSG [8], RI [18], MLRI [19], IRI [2]) due to its iterative processing flow. However, it delivers considerably better demosaicing performance than these fast methods, as described in Sections III-B.1 and III-B.2. On the other hand, our proposed method is significantly more efficient than the LSSC [14] and LDI-NAT [6] methods, where the LSSC method delivers the most comparable demosaicing performance to that of ours in terms of objective and subjective

TABLE IV

AVERAGE RUNNING TIME (IN SECONDS) PER IMAGE ON THE IMAX AND KODAK DATASETS

Methods	IMAX (500×500)	Kodak (512×768)
LSSC [14]	113.5446	173.7845
GBTF [5]	0.0713	0.1214
LDI-NAT [6]	277.1109	439.9941
MSG [8]	4.3644	6.6752
RI [18]	0.5473	0.9066
MLRI [19]	0.8638	1.3556
DDR [17]	4.8126	7.2928
FR [17]	8.5059	13.7943
ARI [21]	24.8025	39.5632
IRI [2]	2.4326	4.4233
IRI-PCR (Ours)	16.9421	26.6163

evaluations. Moreover, the DDR [17] and FR methods achieve promising performance, however these two methods highly depend on the parameter values set for the experiments, since each dataset has its own optimized parameter values. This study has shown that the proposed PCR is able to achieve a good trade-off between the computational complexity and demosaicked image's quality.

C. Default Parameters Setting

In this sub-section, the influences affected by several key parameters are analyzed in our proposed PCR method—i.e., the number of atoms, the number of nearest neighbors, the number of training samples, and the number of iterations. The CPSNR performance of the proposed PCR performed on the IMAX and Kodak datasets are evaluated and presented in this sub-section, while similar conclusions can be drawn on the evaluation of the SSIM results. In order to determine the best default value for each parameter, experimental simulations have been performed by varying the values of the parameter under study, while fixing other parameters unchanged.

1) Influence of the Number of Atoms: In Fig. 7, the first plot shows the average CPSNR (dB) results yielded on the IMAX (blue curve) and Kodak (red curve) by experimenting different values of the atom numbers. It can be observed that by increasing the value of the atom numbers, the CPSNR gain resulted by the proposed PCR becomes larger. However, the gain starts to have slight increment when the atom number value becomes larger than 2^{12} (i.e., 4,096). Therefore, 4,096 is set as the default value for the atom numbers in our experiments.

2) Influence of the Number of Nearest Neighbors: The second column of Fig. 7 presents the changes in CPSNR of the proposed PCR by using various values of the nearest neighbor number. One can see that when the number of nearest neighbors increases from 2^9 to 2^{11}, the CPSNR performance gain is significantly improved, but starts to level off beyond 2^{11}. This implies that larger nearest neighbor number will lead to better demosaicing results and higher performance. However, a larger value of nearest neighbor number will require higher computational complexity on both training and demosaicing stages. Based on the above observation, 2^{11} (i.e., 2,048) is selected as the value of nearest neighbor number in our experiments.

TABLE V

AVERAGE CPSNR RESULTS (IN dB) AND SSIM OF PCR ON THE KODAK AND THE IMAX DATASETS WITH DIFFERENT NUMBERS OF ITERATIONS

Iterations	IMAX		Kodak		IMAX+Kodak	
	CPSNR	SSIM	PSNR	CPSNR	CPSNR	SSIM
1	38.0335	0.9670	40.3675	0.9846	39.3672	0.9771
2	38.1948	0.9672	40.7562	0.9859	39.6585	0.9779
3	38.2563	0.9676	40.7792	0.9860	39.6979	0.9781
4	38.2725	0.9676	40.8279	0.9861	39.7327	0.9782

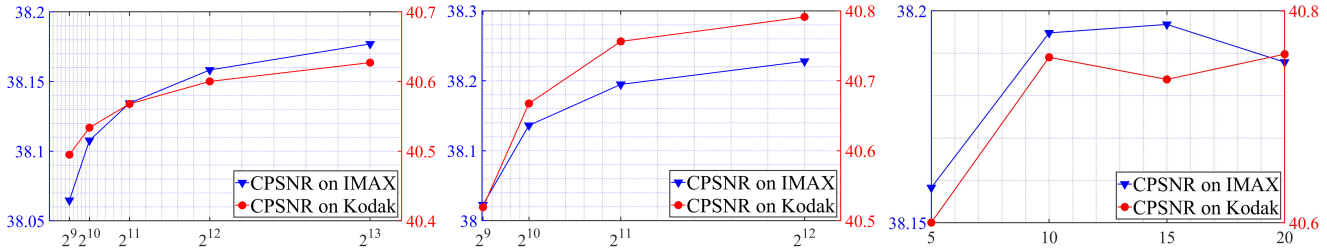
TABLE VI

AVERAGE CPSNR PERFORMANCE COMPARISON ON THE IMAX AND THE KODAK DATASETS UNDER DIFFERENT LEVELS OF GAUSSIAN NOISE POWER

Methods	IMAX				Kodak			
	Noise Free	$\delta=2$	$\delta=6$	$\delta=10$	Noise Free	$\delta=2$	$\delta=6$	$\delta=10$
LSSC [14]	36.1698	34.7432	30.2448	26.9420	**41.4416**	**38.2930**	31.4007	27.3711
GBTF [5]	33.9280	33.2103	30.2077	**27.4680**	40.4225	**38.0958**	**32.1532**	**28.2910**
LDI-NAT [6]	36.2023	35.0200	30.7322	27.4263	37.6720	36.0874	31.0145	27.3524
MSG [8]	34.7212	33.6462	29.7939	26.6936	40.9890	37.8529	31.1500	27.1238
RI [18]	36.4752	35.0750	30.6427	27.3601	38.5400	36.5599	31.0629	27.3503
MLRI [19]	36.6240	35.1359	30.4885	27.0773	39.5600	37.0774	30.9833	27.0610
DDR [17]	37.1475	35.5088	30.6647	27.2026	**41.0680**	37.8947	31.1457	27.1018
FR [17]	**37.4745**	**35.7935**	**30.7877**	27.2942	**41.0392**	**37.8997**	31.1821	27.1400
ARI [21]	**37.5963**	**35.8770**	**31.0424**	**27.5642**	39.9288	37.4264	**31.3896**	**27.4760**
IRI [2]	37.0597	35.4542	30.7813	27.4109	39.7181	37.1718	31.0987	27.2493
IRI-PCR	**38.1948**	**36.3205**	**31.1380**	**27.6131**	40.7562	37.8265	**31.4024**	**27.4868**

3) Influence of the Number of Training Samples: The third column of Fig. 7 shows the average CPSNR results on the combined dataset IMAX+Kodak by using various numbers of training samples. It is easy to figure out that the number of training samples has a weak impact on the performance of the proposed PCR. In addition, unlike the influences introduced by atoms and nearest neighbors, there is a random variation in performance gains incurred by training samples. That is to say, more training samples cannot guarantee higher performance. In our experiments, 10 million is set as the default value as our training samples.

4) Influence of the Number of Iterations: Since the proposed PCR conducts demosaicing on the input image in an iterative way, thus the more the number of iterations, the better the demosaiked results, at the expense of consuming more running time. Table V reports the average CPSNR and SSIM results of PCR on the datasets IMAX and Kodak under different numbers of iterations. As one can see from Table V, both CPSNR and SSIM increase with more iterations. Specifically, the demosaicked results are greatly improved after the second iteration and slightly increased in further iterations. As a result, the iteration number is set at 2 as the default value for our proposed PCR.



(a) Number of Atoms (b) Number of Nearest Neighbors (c) Number of Training Samples ($\times 10^6$)

Fig. 7. A study of the resulted performances in CPSNR of the proposed PCR algorithm with respect to the values set for: (a) the atom number, (b) the nearest neighbor number, and (c) the number of training samples. These simulations are conducted on the datasets IMAX (blue curve) and Kodak (red curve).

D. Noisy Image Demosaicing

It is highly relevant, and interesting, to study the demosaicing of *noisy* images, since such images are often acquired under low-lighting condition, in which a high-valued ISO is normally set in digital camera; consequently, granular noise tends to present in the acquired photo. It is important to note that the granular noise is part of the *ground truth*, and any attempt to remove or reduce such granular noise is out of the demosaicing's objective and falls within the scope of image denoising. Therefore, we shall focus on demosaicing only, without incorporating any denoising functionality in the demosaicing process as some existing works do (e.g., [27]).

To investigate the demosaicing performance of noisy images, various degrees of Gaussian noise are simulated and added on the color images of the IMAX and Kodak datasets. Three levels of noise powers σ^2 (denoted in terms of the standard deviation in the Table VI and Table VII) are experimented for $\sigma = 2$, 6, and 10, separately; this is similar to the works performed in [27] and [28]. Different demosaicing methods are then performed, and the obtained performance measurements are documented in Tables VI for the average CPSNR and Table VII for the average SSIM. In each table, the first-, second-, and third-ranked top performers are boldfaced in red, blue, and black, respectively.

From these two tables, one can observe that our proposed PCR outperforms other methods on all noise levels simulated on the IMAX dataset. For the Kodak dataset, our proposed method also delivers attractive performance that is comparable to two best methods (i.e., LSSC [14] and DDR [17]). Lastly, it is worthwhile to highlight that our proposed PCR can consistently improve the performance of the IRI [2] on all three noise levels experimented on the IMAX and Kodak datasets.

E. Performance Improvement by the PCR Framework

It is important to note that the proposed PCR framework is generic in the sense that any demosaicing method can be straightforwardly incoporated into the proposed PCR framework by simply replacing the IRI depicted in Fig. 1 for boosting its demosaicked image quality. To justify this claim, six demosaicing methods are experimented individually, followed by conducting their performance evaluations to see how much additional performance gain can be offered by the PCR framework. The results are documented in Table VIII.

TABLE VII

AVERAGE SSIM PERFORMANCE COMPARISON ON THE IMAX AND THE KODAK DATASETS UNDER DIFFERENT LEVELS OF GAUSSIAN NOISE POWER

Methods	IMAX				Kodak			
	Noise Free	$\delta=2$	$\delta=6$	$\delta=10$	Noise Free	$\delta=2$	$\delta=6$	$\delta=10$
LSSC [14]	0.9519	0.9228	0.7937	0.6844	**0.9869**	**0.9593**	0.8222	0.7024
GBTF [5]	0.9267	0.9040	0.7816	0.6473	0.9853	**0.9619**	**0.8275**	0.6770
LDI-NAT [6]	0.9543	0.9280	0.8050	0.6998	0.9756	0.9488	0.8169	0.7014
MSG [8]	0.9349	0.9055	0.7787	0.6735	0.9866	0.9555	0.8114	0.6903
RI [18]	0.9597	0.9309	0.8089	0.7057	0.9788	0.9503	0.8185	**0.7047**
MLRI [19]	0.9594	0.9286	0.7987	0.6905	0.9829	0.9523	0.8109	0.6917
DDR [17]	0.9613	0.9309	0.8019	0.6944	**0.9871**	0.9558	0.8109	0.6894
FR [17]	0.9633	0.9331	0.8051	0.6980	**0.9870**	0.9560	0.8125	0.6917
ARI [21]	**0.9658**	**0.9372**	**0.8134**	**0.7105**	0.9819	0.9538	**0.8251**	**0.7129**
IRI [2]	0.9621	0.9332	0.8120	0.7082	0.9820	0.9506	0.8166	0.7023
IRI-PCR	**0.9672**	**0.9388**	**0.8155**	**0.7113**	0.9859	**0.9566**	**0.8235**	**0.7083**

TABLE VIII

AVERAGE CPSNR, SSIM AND MSSIM PERFORMANCE COMPARISON OF DIFFERENT DEMOSAICING METHODS EXPERIMENTED ON THE IMAX AND THE KODAK DATASETS

Methods	IMAX			Kodak		
	CPSNR	SSIM	MSSIM	CPSNR	SSIM	MSSIM
GBTF [5]	33.9280	0.9267	0.9859	40.4225	0.9853	0.9964
GBTF-PCR	37.6574	0.9635	0.9942	40.6003	0.9855	0.9966
MSG [8]	34.7212	0.9349	0.9882	40.9890	0.9866	0.9967
MSG-PCR	37.6588	0.9635	0.9942	41.1311	0.9877	0.9968
MLRI [19]	36.6240	0.9594	0.9935	39.5600	0.9829	0.9959
MLRI-PCR	37.8238	0.9654	0.9946	40.0796	0.9843	0.9962
ARI [21]	37.5963	0.9658	0.9945	39.9288	0.9819	0.9961
ARI-PCR	38.3006	0.9679	0.9951	40.7406	0.9851	0.9966
CDMNet [25]	39.3406	0.9716	0.9958	42.2970	0.9885	0.9975
CDMNet-PCR	39.3507	0.9717	0.9958	42.3107	0.9885	0.9776
IRI [2]	37.0597	0.9621	0.9943	39.7181	0.9820	0.9961
IRI-PCR	38.1948	0.9672	0.9949	40.7562	0.9859	0.9960

One can see that if the demosaicing method is not so state-of-the-art and/or the image to be demosaicked is quite challenging, our proposed PCR is able to enhance the algorithm performance quite effectively. For example, compare

GBTF [5] and GBTF-PCR, one can see that there is about 3.73 dB of the CPSNR gain yielded when the GBTF [5] is incorporated into our PCR framework (i.e., GBTF-PCR). On the other hand, with more advanced demosaicing methods, it is expected that the benefit gained from the proposed PCR will be getting less. For example, the CDMNet [25], which is a state-of-the-art deep-learning approach, can only yield additional 0.01 dB of the CPSNR gain in our PCR framework as shown in Table VIII.

IV. Conclusion

In this paper, a generic color image post-refinement framework is proposed, called the *progressive collaborative representation* (PCR), which is exploited for progressively improving the demosaicked image through multiple stages. The proposed PCR has two phases as follows. In the offline training phase, the more refined demosaicked images in any given stage are the output from the previous stage. The high-frequency features (such as edges) extracted from the difference between these images and their corresponding original full-RGB images are the basis for re-training. The generated projection matrix will be exploited for further refining the demosaicked images of the current stage. These procedures will be repeated in the next stage. The above-mentioned methodology has an effect that any errors or artifacts that have not been sufficiently corrected from the previous stage will have a chance to be further corrected in the current stage. At the end of offline training, all the generated projection matrices will be used in the corresponding stages in the online refinement phase to conduct post-refinement process.

Extensive experiments conducted on two widely used image datasets (i.e., the IMAX and the Kodak) for evaluating the performance of color image demosaicing have clearly shown that our proposed PCR framework is able to effectively address several key adverse factors through a unfied treatment via the generated projection matrices. The above-mentioned key factors spectral correlation existing among three color channels, granular noise incurred in low-lighting conditions (Section III-D), and demosaicing algorithm's deficiency (Section III-E). All these have indicated that the proposed PCR post-refinement framework is able to deliver consistent improvement and makes the developed algorithm's performance more robust against adverse factors.

Acknowledgment

The authors would like to thank Associate Editor Professor Hitoshi Kiya and the three anonymous reviewers for their insightful comments and suggestions on our initial manuscript of this paper.

References

[1] B. E. Bayer, "Color imaging array," U.S. Patent 3 971 065, Jul. 20, 1976.
[2] W. Ye and K.-K. Ma, "Color image demosaicing using iterative residual interpolation," *IEEE Trans. Image Process.*, vol. 24, no. 12, pp. 5879–5891, Dec. 2015.
[3] J. F. Hamilton, Jr., and J. E. Adams, Jr., "Adaptive color plan interpolation in single sensor color electronic camera," U.S. Patent 5 629 734, May 13, 1997.
[4] L. Zhang and X. Wu, "Color demosaicking via directional linear minimum mean square-error estimation," *IEEE Trans. Image Process.*, vol. 14, no. 12, pp. 2167–2178, Dec. 2005.
[5] I. Pekkucuksen and Y. Altunbasak, "Gradient based threshold free color filter array interpolation," in *Proc. IEEE Int. Conf. Image Process.*, Sep. 2010, pp. 137–140.
[6] X. Wu, "Color demosaicking by local directional interpolation and nonlocal adaptive thresholding," *J. Electron. Imag.*, vol. 20, no. 2, Apr. 2011, Art. no. 023016.
[7] I. Pekkucuksen and Y. Altunbasak, "Edge strength filter based color filter array interpolation," *IEEE Trans. Image Process.*, vol. 21, no. 1, pp. 393–397, Jan. 2012.
[8] 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.
[9] X. Li, "Demosaicing by successive approximation," *IEEE Trans. Image Process.*, vol. 14, no. 3, pp. 370–379, Mar. 2005.
[10] 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.
[11] L. Chen, K.-H. Yap, and Y. He, "Subband synthesis for color filter array demosaicking," *IEEE Trans. Syst., Man, Cybern. A, Syst., Humans*, vol. 38, no. 2, pp. 485–492, Mar. 2008.
[12] 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.
[13] L. Fang, O. C. Au, Y. Chen, A. K. Katsaggelos, H. Wang, and X. Wen, "Joint demosaicing and subpixel-based down-sampling for Bayer images: A fast frequency-domain analysis approach," *IEEE Trans. Multimedia*, vol. 14, no. 4, pp. 1359–1369, Aug. 2012.
[14] J. Mairal, F. Bach, J. Ponce, G. Sapiro, and A. Zisserman, "Non-local sparse models for image restoration," in *Proc. IEEE 12th Int. Conf. Comput. Vis.*, vol. 29, Sep. 2009, pp. 54–62.
[15] J. Li, C. Bai, Z. Lin, and J. Yu, "Optimized color filter arrays for sparse representation-based demosaicking," *IEEE Trans. Image Process.*, vol. 26, no. 5, pp. 2381–2393, May 2017.
[16] J. Duran and A. Buades, "Self-similarity and spectral correlation adaptive algorithm for color demosaicking," *IEEE Trans. Image Process.*, vol. 23, no. 9, pp. 4031–4040, Sep. 2014.
[17] J. Wu, R. Timofte, and L. Van Gool, "Demosaicing based on directional difference regression and efficient regression priors," *IEEE Trans. Image Process.*, vol. 25, no. 8, pp. 3862–3874, Aug. 2016.
[18] D. Kiku, Y. Monno, M. Tanaka, and M. Okutomi, "Residual interpolation for color image demosaicking," in *Proc. IEEE Int. Conf. Image Process.*, Sep. 2013, pp. 2304–2308.
[19] D. Kiku, Y. Monno, M. Tanaka, and M. Okutomi, "Minimized-Laplacian residual interpolation for color image demosaicking," *Proc. SPIE*, vol. 9023, Mar. 2014, Art. no. 90230L.
[20] W. Ye and K.-K. Ma, "Image demosaicing by using iterative residual interpolation," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Oct. 2014, pp. 1862–1866.
[21] Y. Monno, D. Kiku, M. Tanaka, and M. Okutomi, "Adaptive residual interpolation for color and multispectral image demosaicking," *Sensors*, vol. 17, no. 12, p. 2787, Dec. 2017.
[22] X. Li, B. Gunturk, and L. Zhang, "Image demosaicing: A systematic survey," *Proc. SPIE*, vol. 6822, Jan. 2008, Art. no. 68221J.
[23] 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.
[24] R. Tan, K. Zhang, W. Zuo, and L. Zhang, "Color image demosaicking via deep residual learning," in *Proc. IEEE Int. Conf. Multimedia Expo (ICME)*, Jul. 2017, pp. 793–798.
[25] K. Cui, Z. Jin, and E. Steinbach, "Color image demosaicking using a 3-Stage convolutional neural network structure," in *Proc. 25th IEEE Int. Conf. Image Process. (ICIP)*, Oct. 2018, pp. 2177–2181.
[26] D. S. Tan, W.-Y. Chen, and K.-L. Hua, "DeepDemosaicking: Adaptive image demosaicking via multiple deep fully convolutional networks," *IEEE Trans. Image Process.*, vol. 27, no. 5, pp. 2408–2419, May 2018.
[27] M. Gharbi, G. Chaurasia, S. Paris, and F. Durand, "Deep joint demosaicking and denoising," *ACM Trans. Graph.*, vol. 35, no. 6, pp. 1–12, Nov. 2016.
[28] F. Kokkinos and S. Lefkimmiatis, "Iterative joint image demosaicking and denoising using a residual denoising network," *IEEE Trans. Image Process.*, vol. 28, no. 8, pp. 4177–4188, Aug. 2019.
[29] B. Zhong, K.-K. Ma, and Z. Lu, "Predictor-corrector image interpolation," *J. Vis. Commun. Image Represent.*, vol. 61, pp. 50–60, May 2019.

[30] R. Zeyde, M. Elad, and M. Protter, "On single image scale-up using sparse-representations," in *Proc. Int. Conf. Curves Surf.*, 2010, pp. 711–730.
[31] L. Zhang, M. Yang, and X. Feng, "Sparse representation or collaborative representation: Which helps face recognition?" in *Proc. Int. Conf. Comput. Vis.*, Nov. 2011, pp. 471–478.
[32] R. Timofte, V. De Smet, and L. Van Gool, "A+: Adjusted anchored neighborhood regression for fast super-resolution," in *Proc. IEEE Asian Conf. Comput. Vis.*, Nov. 2014, pp. 111–126.
[33] F. Zhang, X. Wu, X. Yang, W. Zhang, and L. Zhang, "Robust color demosaicking with adaptation to varying spectral correlations," *IEEE Trans. Image Process.*, vol. 18, no. 12, pp. 2706–2717, Dec. 2009.
[34] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: From error visibility to structural similarity," *IEEE Trans. Image Process.*, vol. 13, no. 4, pp. 600–612, Apr. 2004.

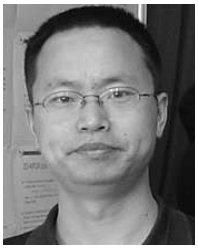
Zhangkai Ni (Student Member, IEEE) received the M.E. degree in communication engineering from the School of Information Science and Engineering, Huaqiao University, Xiamen, China, in 2017. He is currently pursuing the Ph.D. degree with the Department of Computer Science, City University of Hong Kong, Hong Kong. He was a Research Engineer with the School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore, from 2017 to 2018. His current research interests include perceptual image processing, deep learning, and computer vision.

Kai-Kuang Ma (Fellow, IEEE) received the Ph.D. degree in electrical and computer engineering from North Carolina State University, Raleigh, NC, USA. He was a Member of Technical Staff with the Institute of Microelectronics (IME), from 1992 to 1995, working on digital video coding and the MPEG standards. From 1984 to 1992, he was with IBM Corporation, Kingston, NY, USA, and then Research Triangle Park, NC, USA, engaging on various DSP and VLSI advanced product development. He is currently a Professor with the School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore. He has published extensively in well-known international journals, conferences, and MPEG standardization meetings. He holds one USA patent on fast motion estimation algorithms. His research interests are in the areas of fundamental image/video processing and applied computer vision. He was serving as the Singapore MPEG Chairman and the Head of Delegation from 1997 to 2001. On the MPEG contributions, two fast motion estimation algorithms (Diamond Search and MVFAST) produced from his research group have been adopted by the MPEG-4 standard, as the reference core technology for fast motion estimation.

He is an Elected Member of three IEEE Technical Committees including the Image and Multidimensional Signal Processing (IMDSP) Committee, the Multimedia Communications Committee, and Digital Signal Processing. He was the Chairman of IEEE Signal Processing, Singapore Chapter from 2000 to 2002. He is the General Chair of organizing a series of international standard meetings (MPEG and JPEG), JPEG2000, and MPEG-7 workshops held in Singapore, March 2001. He has served various roles in professional societies including the General Co-Chair of APSIPA-2017, ISPACS-2017, ACCV-2016 (Workshop), VCIP-2013, Technical Program Co-Chair of ICASSP-2022, ICIP-2004, ISPACS-2007, IIH-MSP-2009, and PSIVT-2010; and Area Chair of ACCV-2009 and ACCV-2010. He also had extensive editorship contributions in several international journals including a Senior Area Editor since 2015 and an Associate Editor from 2007 to 2010 of the IEEE TRANSACTIONS ON IMAGE PROCESSING, an Associate Editor since 2015 of the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY, an Associate Editor from 2014 to 2016 of the IEEE SIGNAL PROCESSING LETTERS, an Editor from 1997 to 2012 of the IEEE TRANSACTIONS ON COMMUNICATIONS, an Associate Editor from 2002 to 2009 of the IEEE TRANSACTIONS ON MULTIMEDIA, and an Editorial Board Member from 2005 to 2014 of the *Journal of Visual Communication and Image Representation*. He was elected as a Distinguished Lecturer of the IEEE Circuits and Systems Society from 2008 to 2009.

Huanqiang Zeng (Senior Member, IEEE) received the B.S. and M.S. degrees in electrical engineering from Huaqiao University, Xiamen, China, and the Ph.D. degree in electrical engineering from Nanyang Technological University, Singapore.

He is currently a Full Professor with the School of Information Science and Engineering, Huaqiao University, Xiamen, China. He was a Postdoctoral Fellow with The Chinese University of Hong Kong, Hong Kong. He has published more than 100 articles in well-known journals and conferences, including IEEE TIP, TCSVT, TITS, and three best poster/article awards (in International Forum of Digital TV and Multimedia Communication 2018 and in Chinese Conference on Signal Processing from 2017 to 2019. His research interests include image processing, video coding, machine learning, and computer vision. He has also been actively serving as the General Co-Chair for the IEEE International Symposium on Intelligent Signal Processing and Communication Systems 2017 (ISPACS2017), the Co-Organizers of ICME2020 Workshop on 3D Point Cloud Processing, Analysis, Compression, and Communication, the Technical Program Co-Chair for the Asia-Pacific Signal and Information Processing Association Annual Summit and Conference 2017 (APSIPA-ASC2017) and ISPACS2019, the, the Area Chair for the IEEE International Conference on Visual Communications and Image Processing (VCIP2015), the Technical Program Committee Member for multiple flagship international conferences, and the Reviewer for numerous international journals and conferences. He has been actively serving as the Associate Editor for the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY, IEEE ACCESS, and *IET Electronics Letters*, the Guest Editor for the *Journal of Visual Communication and Image Representation, Multimedia Tools and Applications*, and the *Journal of Ambient Intelligence and Humanized Computing*.



Baojiang Zhong (Senior Member, IEEE) received the B.S. degree in mathematics from Nanjing Normal University, China, in 1995, the M.S. degree in mathematics from the Nanjing University of Aeronautics and Astronautics (NUAA), China, in 1998, and the Ph.D. degree in mechanical and electrical engineering from NUAA, China, in 2006. From 1998 to 2009, he was on the Faculty of the Department of Mathematics, NUAA, where he was an Associate Professor. In 2009, he joined the School of Computer Science and Technology, Soochow University, China, where he is currently a Full Professor. His research interests include computer vision, image processing, and numerical linear algebra.