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 (DLMME) 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...
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 demosaiced 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 demosaiced 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 demosaiced 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 demosaiced images obtained from the previous stage. The refined demosaiced 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 demosaiced 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 $i = 1, 2, \ldots, NO$), the simulated mosaicked images $M_i$, and their initially-generated demosaicked images $D^{(1)}_i$ are prepared for the training stage, from which the original $O_i$ and the demosaicked images $D^{(1)}_i$ 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 $D^{(1)}_i$ of the current (first) stage; thus, images $D^{(2)}_i$ are further refined and denoted as $D^{(2)}_i$, 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 $D^{(1)}_i$, 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...
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 segmenting the filtered images into $3 \times 3$ 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 $i = 1, 2, \ldots, N_f$) 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:

$$\left[\Xi^{(1)},\{c_i\}\right] = \arg \min_{[\Xi^{(1)},\{c_i\}]} \sum_i \left\|y_i^{(1)} - \Xi^{(1)} \cdot c_i\right\|^2,$$

subject to $\|c_i\|_0 \leq L$ and $i = 1, 2, \ldots, N_f$, \hspace{1cm} (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.

$$\left[\Xi^{(1)},\{c_i\}\right] = \arg \min_{[\Xi^{(1)},\{c_i\}]} \sum_i \left\|x_i^{(1)} - \Xi^{(1)} \cdot c_i\right\|^2,$$

subject to $\|c_i\|_0 \leq L$ and $i = 1, 2, \ldots, N_f$, \hspace{1cm} (1)
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 \( \{f_{D_k}^{(1)}, f_{E_k}^{(1)}\} \). Here, the absolute value of the dot product between the atom \( d_k^{(1)} \) and each individual feature patch in \( f_{D_k}^{(1)} \) will be computed to measure the degree of similarity; that is,

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

(2)

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

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

(3)

where \( N_{D_k}^{(1)} \) is the identified neighborhood from the demosaicked image, \( \omega \) is a regularization coefficient of \( x_i^{(1)} \) over \( N_{D_k}^{(1)} \), and \( \lambda \) is used to alleviate the ill-posed problem. The above-stated equation has a closed-form solution, which is

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

(4)

where \( I \) is the identity matrix.

The demosaicked image patches can be refined by applying the same coefficients \( \omega \) 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 \( \omega \) 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:

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

(5)

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

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

(6)

The projection matrix \( P^{(1)} \) is composed of matrices \( p_k^{(1)} \) for \( k = 1, 2, \ldots, N_d \) and used to update \( D^{(1)} \) (as the input) for yielding a refined \( D^{(2)} \) (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.

\section*{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

\[
f_{D_i}^{(n+1)} = p_k^{(n)} \cdot f_{D_i}^{(n)}.
\]  

(7)

The \textit{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

\begin{algorithm}
\SetAlgoLined
\KwResult{A set of full-color training images \( O_i \) (where \( i = 1, 2, \ldots, N_O \)).}
\KwResult{The projection matrices \( P^{(n)} \) (where \( n = 1, 2, \ldots, N \)).}
\KwFor{\( i = 1, 2, \ldots, N_O \), \( n = 1, 2, \ldots, N \)}{Compute the Projection Matrices via Offline Training.
\SetKwProg{Kw}{Steps:}{:}{end}
\KwInput{Training the dictionary \( \Xi^{(n)} \) based on \( y_i^{(n)} \) using the K-SVD method via (1);}
\KwSearch{Extract feature patches \( f_{D_i}^{(n)} \) 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 \( i = 1, 2, \ldots, N_d \)).}
\KwSearch{Train the dictionary \( \Xi^{(n)} \) based on \( y_i^{(n)} \) using the K-SVD method via (1);}
\KwSearch{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.}
\KwSearch{Search the corresponding \( N_{E_i}^{(n)} \) from \( f_{E_i}^{(n)} \);}
\KwSearch{Search the corresponding \( N_{D_k}^{(n)} \) from \( f_{D_k}^{(n)} \);}
\KwSearch{Compute the projection matrix \( P^{(n)} \) via (6).}
\KwUpdate{Update \( D_i^{(n)} \) to \( D_i^{(n+1)} \) via the projection matrix \( P^{(n)} \).}
\KwOutput{Output the projection matrices \( P^{(n)} \);}
\end{algorithm}
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 $z_i^{(n+1)}$ 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,

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

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 the 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 among three color channels and is 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.

$$\text{CMSE} = \frac{\sum_{C \in \{R,G,B\}} \sum_{H} \sum_{W} \| I_C^o(i, j) - I_C^I(i, j) \|^2}{3 \times H \times W},$$  \(9\)

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 $I_C^o(i, j)$ and $I_C^I(i, j)$ 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 demosaiced images. The SSIM considers the degradations incurred on the demosaiced 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 demosaiced 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 $3 \times 3$ pixels with an overlap of 2 pixels between adjacent patches. Four 1D high-pass filters (i.e., $f_1 = [-1, -1, 0, 1]$, $f_2 = [-1, 0, 1]^T$, $f_3 = [1, 0, -2, 0, 1]$, 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 demosaiced 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.

<table>
<thead>
<tr>
<th>Methods</th>
<th>IMAX</th>
<th>Kodak</th>
<th>IMAX+Kodak</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>PSNR</td>
<td>CPSNR</td>
<td>PSNR</td>
</tr>
<tr>
<td></td>
<td>R</td>
<td>G</td>
<td>B</td>
</tr>
<tr>
<td>LSSC [14]</td>
<td>36.0412</td>
<td>38.8235</td>
<td>34.7159</td>
</tr>
<tr>
<td>MSG [8]</td>
<td>34.3785</td>
<td>37.6532</td>
<td>33.3853</td>
</tr>
<tr>
<td>RI [18]</td>
<td>36.0722</td>
<td>39.9923</td>
<td>35.3501</td>
</tr>
<tr>
<td>DDR [17]</td>
<td>37.0884</td>
<td>40.3345</td>
<td>35.6214</td>
</tr>
<tr>
<td>FR [17]</td>
<td>37.4748</td>
<td>41.0044</td>
<td>35.8077</td>
</tr>
<tr>
<td>IRI-PCR (Ours)</td>
<td>38.2977</td>
<td>41.4566</td>
<td>36.4841</td>
</tr>
</tbody>
</table>

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
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

TABLE II

<table>
<thead>
<tr>
<th>Methods</th>
<th>IMAX</th>
<th>Kodak</th>
<th>IMAX+Kodak</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>SSIM</td>
<td></td>
<td></td>
</tr>
<tr>
<td>LSSC [14]</td>
<td>0.9550</td>
<td>0.9734</td>
<td>0.9274</td>
</tr>
<tr>
<td>GBTF [5]</td>
<td>0.9317</td>
<td>0.9558</td>
<td>0.8927</td>
</tr>
<tr>
<td>LDENAT [6]</td>
<td>0.9601</td>
<td>0.9785</td>
<td>0.9245</td>
</tr>
<tr>
<td>MSG [8]</td>
<td>0.9392</td>
<td>0.9905</td>
<td>0.9025</td>
</tr>
<tr>
<td>RI [18]</td>
<td>0.9592</td>
<td>0.9798</td>
<td>0.9400</td>
</tr>
<tr>
<td>MLRI [19]</td>
<td>0.9605</td>
<td>0.9788</td>
<td>0.9389</td>
</tr>
<tr>
<td>DDR [17]</td>
<td>0.9636</td>
<td>0.9816</td>
<td>0.9387</td>
</tr>
<tr>
<td>FR [17]</td>
<td>0.9659</td>
<td>0.9832</td>
<td>0.9407</td>
</tr>
<tr>
<td>ARI [21]</td>
<td>0.9680</td>
<td>0.9829</td>
<td>0.9467</td>
</tr>
<tr>
<td>IRI [2]</td>
<td>0.9633</td>
<td>0.9818</td>
<td>0.9411</td>
</tr>
<tr>
<td>IRI-PCR (Ours)</td>
<td>0.9703</td>
<td>0.9843</td>
<td>0.9474</td>
</tr>
</tbody>
</table>

TABLE III

<table>
<thead>
<tr>
<th>Methods</th>
<th>IMAX</th>
<th>Kodak</th>
<th>IMAX+Kodak</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>MSSIM</td>
<td></td>
<td></td>
</tr>
<tr>
<td>LSSC [14]</td>
<td>0.9926</td>
<td>0.9969</td>
<td>0.9869</td>
</tr>
<tr>
<td>GBTF [5]</td>
<td>0.9872</td>
<td>0.9942</td>
<td>0.9763</td>
</tr>
<tr>
<td>LDENAT [6]</td>
<td>0.9933</td>
<td>0.9976</td>
<td>0.9868</td>
</tr>
<tr>
<td>MSG [8]</td>
<td>0.9892</td>
<td>0.9954</td>
<td>0.9800</td>
</tr>
<tr>
<td>RI [18]</td>
<td>0.9934</td>
<td>0.9977</td>
<td>0.9899</td>
</tr>
<tr>
<td>MLRI [19]</td>
<td>0.9934</td>
<td>0.9976</td>
<td>0.9895</td>
</tr>
<tr>
<td>DDR [17]</td>
<td>0.9943</td>
<td>0.9979</td>
<td>0.9897</td>
</tr>
<tr>
<td>FR [17]</td>
<td>0.9946</td>
<td>0.9981</td>
<td>0.9900</td>
</tr>
<tr>
<td>ARI [21]</td>
<td>0.9951</td>
<td>0.9980</td>
<td>0.9909</td>
</tr>
<tr>
<td>IRI [2]</td>
<td>0.9945</td>
<td>0.9979</td>
<td>0.9903</td>
</tr>
<tr>
<td>IRI-PCR (Ours)</td>
<td>0.9956</td>
<td>0.9983</td>
<td>0.9912</td>
</tr>
</tbody>
</table>

Authorized licensed use limited to: CITY UNIV OF HONG KONG. Downloaded on March 14,2020 at 09:27:29 UTC from IEEE Xplore. Restrictions apply.
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.
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.

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
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.

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 demosaikmed 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 demosaikmed 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.
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 $\sigma^2$ (denoted in terms of the standard deviation in the Table VI and Table VII) are experimented for $\sigma = 2, 6, \text{and } 10$, 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 incorporated 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.

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 CDNet [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 unified 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

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 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, APSACS-2017, ACCV-2016 (Workshop), VCIP-2013, Technical Program Co-Chair of ICASSP-2022, ICIP-2004, ISPADS-2007, IEE-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.

Huangjiang 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 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.