A Lossless Compression Scheme for Bayer Color Filter Array Images

King-Hong Chung and Yuk-Hee Chan, *Member, IEEE*

***Abstract*—In most digital cameras, Bayer color filter array (CFA) images are captured and demosaicing is generally carried out before compression. Recently, it was found that compression-first schemes outperform the conventional demosaicing-first schemes in terms of output image quality. An efficient prediction-based lossless compression scheme for Bayer CFA images is proposed in this paper. It exploits a context matching technique to rank the neighboring pixels when predicting a pixel, an adaptive color difference estimation scheme to remove the color spectral redundancy when handling red and blue samples, and an adaptive codeword generation technique to adjust the divisor of Rice code for encoding the prediction residues. Simulation results show that the proposed compression scheme can achieve a better compression performance than conventional lossless CFA image coding schemes.**

***Index Terms*—Bayer pattern, color filter array (CFA), digital camera, entropy coding, image compression.**

I. Introduction

TO reduce cost, most digital cameras use a single image sensor to capture color images. A Bayer color filter array (CFA) [1], [2], as shown in Fig. 1, is usually coated over the sensor in these cameras to record only one of the three color components at each pixel location. The resultant image is referred to as a CFA image in this paper hereafter.

In general, a CFA image is first interpolated via a demosaicing process [3]–[9] to form a full color image before being compressed for storage. Fig. 2(a) shows the workflow of this imaging chain.

Recently, some reports [10]–[14] indicated that such a demosaicing-first processing sequence was inefficient in a way that the demosaicing process always introduced some redundancy which should eventually be removed in the following compression step. As a result, an alternative processing sequence [10]–[13] which carries out compression before demosaicing as shown in Fig. 2(b) has been proposed lately. Under this new strategy, digital cameras can have a simpler design and lower power consumption as computationally heavy processes

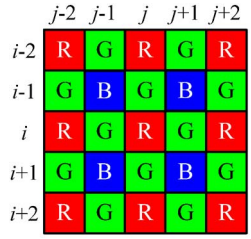

Fig. 1. Bayer pattern having a red sample as its center.

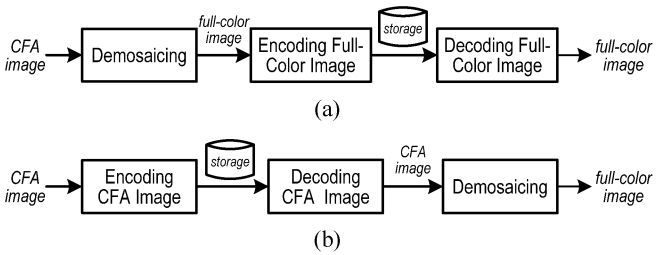

Fig. 2. Single-sensor camera imaging chain: (a) the demosaicing- first scheme; (b) the compression-first scheme.

like demosaicing can be carried out in an offline powerful personal computer. This motivates the demand of CFA image compression schemes.

There are two categories of CFA image compression schemes: lossy and lossless. Lossy schemes compress a CFA image by discarding its visually redundant information. These schemes usually yield a higher compression ratio as compared with the lossless schemes. Schemes presented in [10]–[20] are some examples of this approach. In these schemes, different lossy compression techniques such as discrete cosine transform [15], vector quantization [16], [17] subband coding with symmetric short kernel filters [10], transform followed by JPEG or JPEG 2000 [12], [13], [18]–[20], and low-pass filtering followed by JPEG-LS or JPEG 2000 (lossless mode) [11] are used to reduce data redundancy.

In some high-end photography applications such as commercial poster production, original CFA images are required for producing high quality full color images directly. In such cases, lossless compression of CFA images is necessary. Some lossless image compression schemes like JPEG-LS [21] and JPEG2000 [22] can be used to encode a CFA image but only a fair performance can be attained. Recently, an advanced lossless CFA image compression scheme (LCMI) [23] was proposed. In this scheme, the mosaic data is de-correlated by the Mallat wavelet packet transform, and the coefficients are then compressed by Rice code.

Manuscript received May 17, 2007; revised October 31, 2007. This work was supported by a Grant from the Research Grants Council of the Hong Kong SAR (PolyU 5205/04E) and a Grant from the Hong Kong Polytechnic University (G-U413). The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Bruno Carpentieri.

The authors are with Centre for Multimedia Signal Processing, Department of Electronic and Information Engineering, The Hong Kong Polytechnic University, Hong Kong (e-mail: king.hong.chung@polyu.edu.hk; enyhchan@polyu.edu.hk).

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

Digital Object Identifier 10.1109/TIP.2007.914153

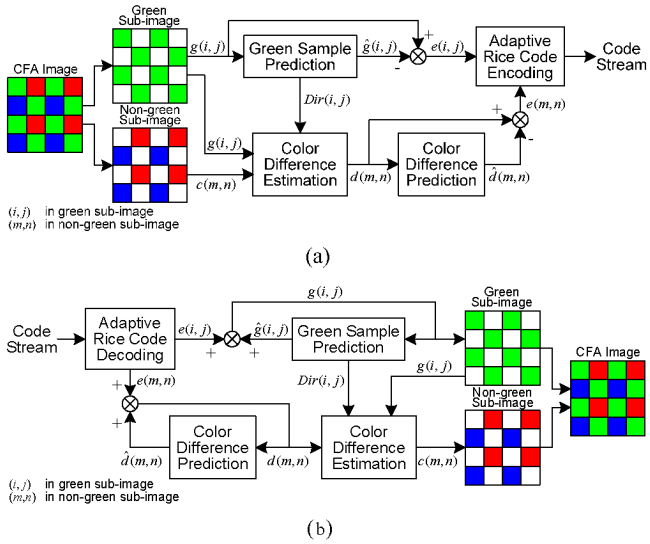

Fig. 3. Structure of the proposed compression scheme: (a) encoder and (b) decoder.

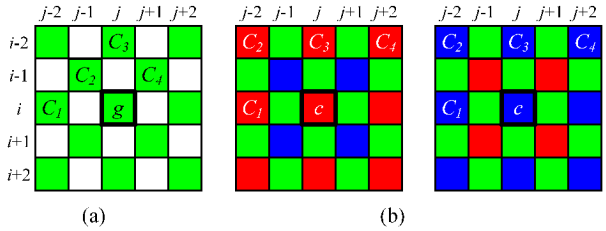

Fig. 4. Positions of the pixels included in the candidate set of (a) a green sample and (b) a red/blue sample.

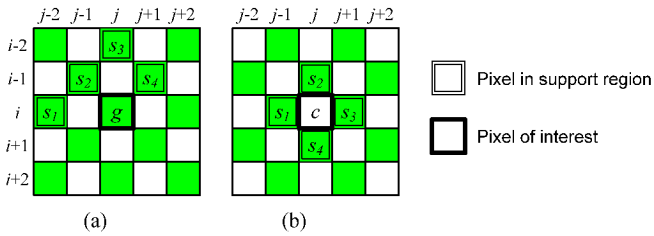

Fig. 5. The support region of (a) a green sample and (b) a red/blue sample.

In this paper, a prediction-based lossless CFA compression scheme as shown in Fig. 3 is proposed. It divides a CFA image into two subimages: a green subimage which contains all green samples of the CFA image and a nongreen subimage which holds the red and the blue samples. The green subimage is coded first and the nongreen subimage follows based on the green subimage as a reference. To reduce the spectral redundancy, the nongreen subimage is processed in the color difference domain whereas the green subimage is processed in the intensity domain as a reference for the color difference content of the nongreen subimage. Both subimages are processed in raster scan sequence with our proposed context matching based prediction technique to remove the spatial dependency. The prediction residue planes of the two subimages are then entropy encoded sequentially with our proposed realization scheme of adaptive Rice code.

Experimental results show that the proposed compression scheme can effectively and efficiently reduce the redundancy in both spatial and color spectral domains. As compared with the existing lossless CFA image coding schemes such as [10]–[12], the proposed scheme provides the best compression performance in our simulation study.

This paper is structured as follows. The proposed context matching based prediction technique is presented in Section II. Section III shows how to estimate a missing green sample in the nongreen subimage of a CFA image for extracting the color difference information when compressing the nongreen subimage. In Section IV, how the prediction residue is adaptively encoded with Rice Code is provided. Section V demonstrates some simulation results, and, finally, a conclusion is given in Section VI.

II. Context Matching Based Prediction

The proposed prediction technique handles the green plane and the nongreen plane separately in a raster scan manner. It weights the neighboring samples such that the one has higher context similarity to that of the current sample contributes more to the current prediction. Accordingly, this prediction technique is referred to as *context matching based prediction* (CMBP) in this paper.

The green plane (green subimage) is handled first as a CFA image contains double number of green samples to that of red/blue samples and the correlation among green samples can be exploited easily as compared with that among red or blue samples. Accordingly, the green plane can be used as a good reference to estimate the color difference of a red or blue sample when handling the nongreen plane (nongreen subimage).

A. Prediction on the Green Plane

As the green plane is raster scanned during the prediction and all prediction errors are recorded, all processed green samples are known and can be exploited in the prediction of the pixels which have not yet been processed.

Assume that we are now processing a particular green sample $g(i,j)$ as shown in Fig. 4(a). The four nearest processed neighboring green samples of $g(i,j)$ form a candidate set $\Phi_{g(i,j)} = \{g(i,j-2), g(i-1,j-1), g(i-2,j), g(i-1,j+1)\}$. The candidates are ranked by comparing their support regions (i.e., context) with that of $g(i,j)$.

The support region of a green sample at position (p,q), $S_{g(p,q)}$, is defined as shown in Fig. 5(a). In formulation, we have

$$S_{g(p,q)} = \{(p,q-2), (p-1,q-1), (p-2,q), (p-1,q+1)\}.$$

The matching extent of the support region of $g(i,j)$ and the support region of $g(m,n)$ for $g(m,n) \in \Phi_{g(i,j)}$ is then measured by

$$\begin{aligned} D\left(S_{g(i,j)}, S_{g(m,n)}\right) &= \left|g_{(i,j-2)} - g_{(m,n-2)}\right| \\ &\quad + \left|g_{(i-1,j-1)} - g_{(m-1,n-1)}\right| \\ &\quad + \left|g_{(i-2,j)} - g_{(m-2,n)}\right| \\ &\quad + \left|g_{(i-1,j+1)} - g_{(m-1,n+1)}\right|. \end{aligned} \quad (1)$$

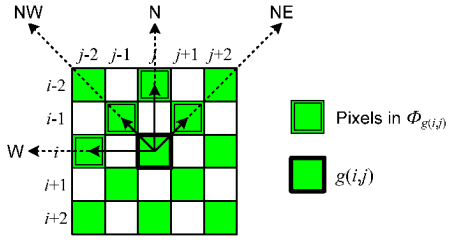

Fig. 6. Four possible directions associated with a green pixel.

Though a higher order distance, such as Euclidian distance, can be used instead of (1) to achieve a better matching performance, we found in our simulations that the improvement was not significant enough to compensate for its high realization complexity.

Let $g(m_k, n_k) \in \Phi_{g(i,j)}$ for $k = 1, 2, 3, 4$ be the four ranked candidates of sample $g(i, j)$ such that $D(S_{g(i,j)}, S_{g(m_u,n_u)}) \leq D(S_{g(i,j)}, S_{g(m_v,n_v)})$ for $1 \leq u < v \leq 4$. The value of $g(i, j)$ can then be predicted with a prediction filter as

$$\hat{g}(i,j) = \text{round}\left(\sum_{k=1}^{4} w_k g(m_k, n_k)\right) \tag{2}$$

where w_k for $k = 1, 2, 3, 4$ are normalized weighting coefficients such that $\sum_{k=1}^{4} w_k = 1$.

Let $\text{Dir}(i, j) \in \{W, NW, N, NE\}$ be a direction vector associated with sample $g(i, j)$. It is defined as the direction pointed from sample $g(i, j)$ to $g(i, j)$'s 1st ranked candidate $g(m_1, n_1)$. Fig. 6 shows all its possible values. This definition applies to all green samples in the green subimage. As an example, Fig. 7 shows the direction map of a testing image shown in Fig. 8. If the direction of $g(i, j)$ is identical to the directions of all green samples in $S_{g(i,j)}$, pixel (i, j) will be considered in a homogenous region and $\hat{g}(i, j)$ will then be estimated to be $g(m_1, n_1)$ directly. In formulation, we have

$$\begin{aligned} \hat{g}(i,j) &= g(m_1, n_1) \text{ if } \text{Dir}(i,j) \\ &= \text{Dir}(a,b)\ \forall (a,b) \in S_{g(i,j)} \end{aligned} \tag{3}$$

which implies $\{w_1, w_2, w_3, w_4\} = \{1, 0, 0, 0\}$. Otherwise, $g(i, j)$ is considered to be in a heterogeneous region and a predefined prediction filter is used to estimate $g(i, j)$ with (2) instead.

In our study, w_k are obtained by quantizing the training result derived by linear regression with a set of training images covering half of the test images shown in Fig. 8. They are quantized to reduce the realization effort of (2). Afterall, when $g(i, j)$ is not in a homogeneous region, the coefficients of the prediction filter used to obtain the result presented in this paper are given by $\{w_1, w_2, w_3, w_4\} = \{5/8, 2/8, 1/8, 0\}$, which allows the realization of (2) to be achieved with only shift and addition operations as shown in (4), at the bottom of the page.

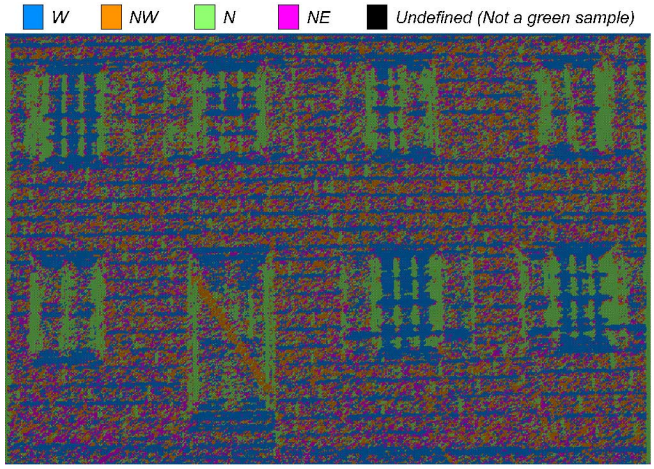

Fig. 7. Direction map of testing image 1.

Fig. 8. Twenty-four digital color images (referred to as image 1 to image 24, from top-to-bottom and left-to-right).

The prediction error is determined with $g(i, j) - \hat{g}(i, j)$. Fig. 9 summaries how to generate the prediction residue of the green plane of a CFA image.

In CMBP, a green sample is classified according to the homogeneity of its local region to improve the prediction performance. Fig. 10 shows the effect of this classification step. By comparing Fig. 10(a) and (b), one can see that the approach with classification can handle the edge regions more effectively and more edge details can be eliminated in the corresponding prediction residue planes. Another supporting observation is the stronger decorrelation power of the approach using classification. Fig. 11 shows the correlation among prediction residues in the green plane of testing image 8 under the two different conditions. The correlation of the residues obtained with region classification is lower, which implies that the approach is more effective in data compression. Besides, the entropy of the prediction residues obtained with region classification is also lower.

$$\hat{g}(i,j) = \text{round}\left(\frac{4g(m_1,n_1) + g(m_1,n_1) + 2g(m_2,n_2) + g(m_3,n_3)}{8}\right) \tag{4}$$

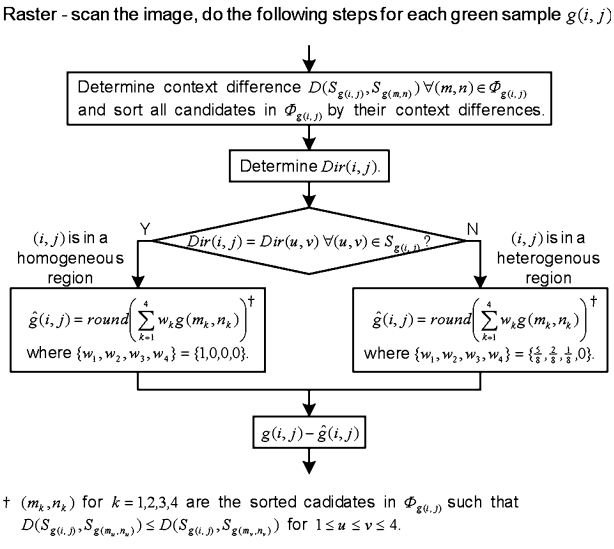

Fig. 9. How to handle the green plane of a CFA image in CMBP.

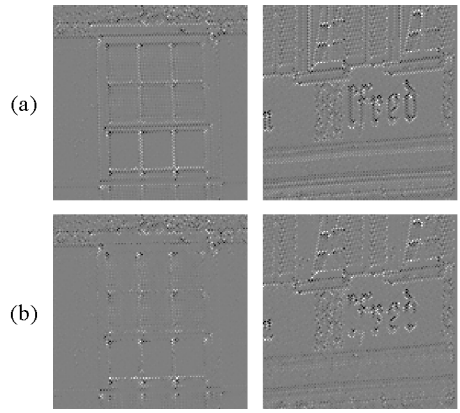

Fig. 10. Prediction residues of the green planes of testing images 1 and 8 (a) without region classification and (b) with region classification.

As far as testing image 8 is concerned, their zero-order entropy values are, respectively, 6.195 and 6.039 bpp.

B. Prediction on the Nongreen Plane

As for the case when the sample being processed is a red or blue sample in the nongreen plane, the prediction is carried out in the color difference domain instead of the intensity domain as in the green plane. This is done to remove the interchannel redundancy.

Since the nongreen plane is processed after the green plane, all green samples in a CFA image are known and can be exploited when processing the nongreen plane. Besides, as the nongreen plane is raster scanned in the prediction, the color difference values of all processed nongreen samples in the CFA image should also be known and, hence, can be exploited when predicting the color difference of a particular nongreen sample.

Let $d(p,q)$ be the green-red (or green-blue) color difference value of a nongreen sample $c(p,q)$. Its determination will be discussed in detail in Section III. For any nongreen sample $c(i,j)$, its candidate set is $\Phi_{c(i,j)} = \{d(i,j-2), d(i-2,j-2), d(i-2,j), d(i-2,j+2)\}$, and its support region (context) is defined as $S_{c(i,j)} = \{(i,j-1),(i-1,j),(i,j+1),(i+1,j)\}$. Figs. 4(b) and Fig. 5(b), show, respectively, the positions of the pixels involved in the definition of $\Phi_{c(i,j)}$ and $S_{c(i,j)}$.

The prediction for a nongreen sample is carried out in the color difference domain. Specifically, the predicted color difference value of sample $c(i,j)$ is given by

$$\hat{d}(i,j) = \text{round}\left(\sum_{k=1}^{4} w_k d(m_k,n_k)\right) \tag{5}$$

where w_k and $d(m_k,n_k)$ are, respectively, the kth predictor coefficient and the kth ranked candidate in $\Phi_{c(i,j)}$ such that $D(S_{c(i,j)}, S_{c(m_u,n_u)}) \le D(S_{c(i,j)}, S_{c(m_v,n_v)})$ for $1 \le u < v \le 4$, where

$$\begin{aligned} &D(S_{c(i,j)}, S_{c(m,n)}) \\ &\quad = |g(i,j-1) - g(m,n-1)| \\ &\qquad + |g(i,j+1) - g(m,n+1)| \\ &\qquad + |g(i-1,j) - g(m-1,n)| \\ &\qquad + |g(i+1,j) - g(m+1,n)|. \end{aligned} \tag{6}$$

In the prediction carried out in the green plane, region homogeneity is exploited to simplify the prediction filter and improve the prediction result. Theoretically, similar idea can be adopted in handling a nongreen sample by considering the direction information of its neighboring samples. For any nongreen sample $c(i,j)$, if the directions of all green samples in $S_{c(i,j)}$ are identical, pixel (i,j) can also be considered as in a homogenous region. Its predicted color difference value $\hat{d}(i,j)$ can then be estimated as shown in (7), at the bottom of the page.

However, such an arrangement is abandoned when a nongreen sample is processed in CMBP as edges are generally deemphasized in the color difference domain. As a matter of fact, simulation results showed that this arrangement did not improve the prediction result of $d(i,j)$. For example, as far as testing image 8 is concerned, the zero-order entropy value of $\{d(i,j) - \hat{d}(i,j) \,|\, (i,j) \in \text{nongreen subimage}\}$ obtained without region classification and that obtained with region

$$\hat{d}(i,j) = \begin{cases} d(i,j-2), & \text{if } \text{Dir}(m,n) = W \\ d(i-2,j-2), & \text{if } \text{Dir}(m,n) = NW \\ d(i-2,j), & \text{if } \text{Dir}(m,n) = N \\ d(i-2,j+2), & \text{if } \text{Dir}(m,n) = NE \end{cases} \quad \forall (m,n) \in S_c(i,j) \tag{7}$$

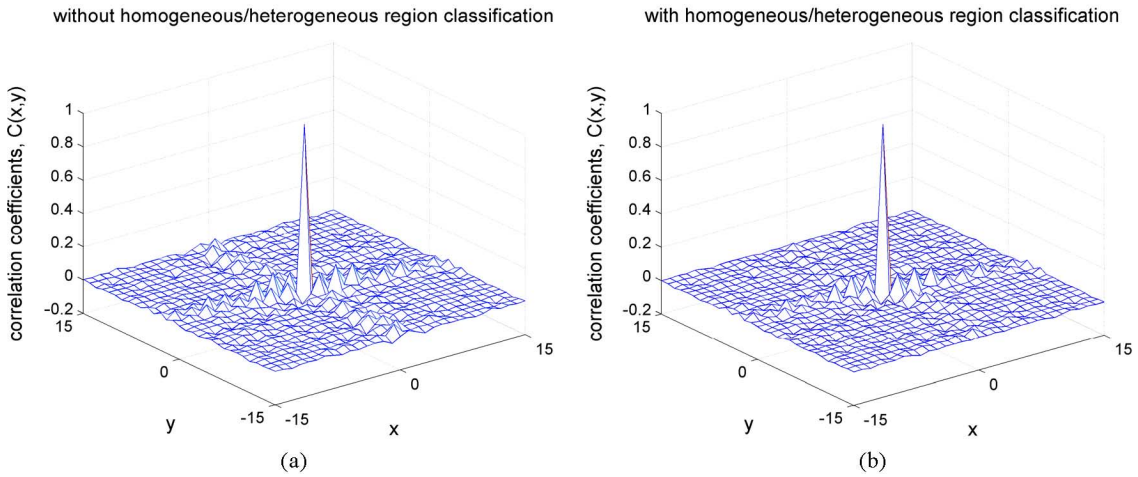

Fig. 11. Correlation among the prediction residues associated with the green subimage of testing image 8 (a) without region classification and (b) with region classification.

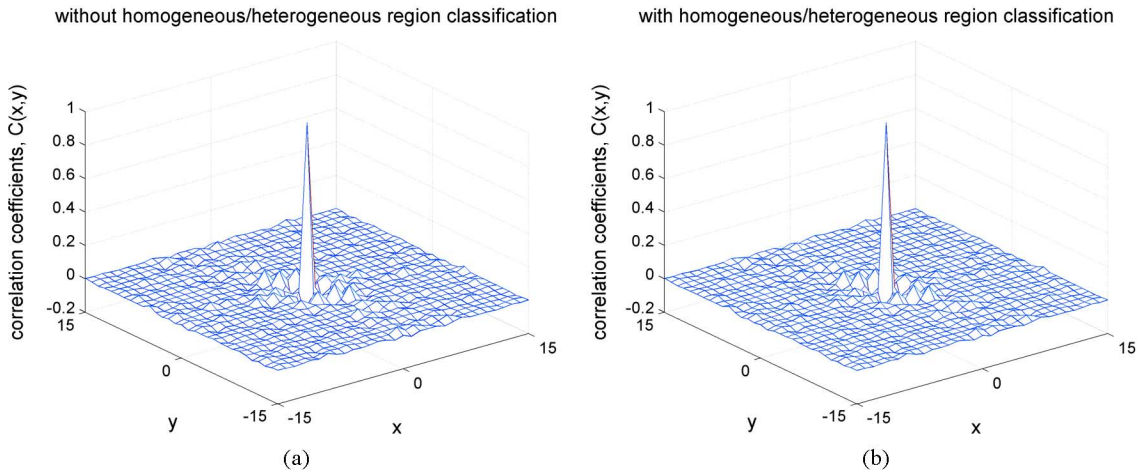

Fig. 12. Correlation among the prediction residues associated with the nongreen subimage of testing image 8: (a) without region classification and (b) with region classification.

classification are, respectively, 5.423 and 5.434 bpp. The entropy of the resultant residue plane is even higher when region classification is exploited. Furthermore, as shown in Fig. 12, the correlation coefficients of the prediction residues are more or less the same no matter whether region classification is used or not, which shows that region classification does not effectively contribute to the decorrelation performance. As a result, in the proposed scheme, a single predefined prediction filter is used to estimate $d(i,j)$ with (5) no matter whether the pixel is in a homogeneous region.

Again, w_k are trained with the same set of training images used to train the predictor coefficients in (2). For the compression results reported in this paper, the predictor used for the color difference prediction is shown in (8), at the bottom of the page.

The prediction error is then obtained with $d(i,j) - \hat{d}(i,j)$. Fig. 13 summaries how to generate the prediction residue of the corresponding color-difference plane for the nongreen plane of a CFA image.

In CMBP, all real green, red, and blue samples are encoded in a raster scan manner. The four samples used for predicting sample $g(i,j)$ in (2) are $g(i,j)$'s closest processed neighboring samples of the same color. They have the highest correlation to $g(i,j)$ in different directions and, hence, can provide a good prediction result even in an edge region. A similar argument applies to explain why $\Phi_{c(i,j)}$ is used when handling a nongreen sample $c(i,j)$.

As for the support region, no matter the concerned sample is green or not, its support is defined based on its four closest

$$\hat{d}(i,j) = \text{round}\left(\frac{4d(m_1,n_1) + 2d(m_2,n_2) + d(m_3,n_3) + d(m_4,n_4)}{8}\right) \tag{8}$$

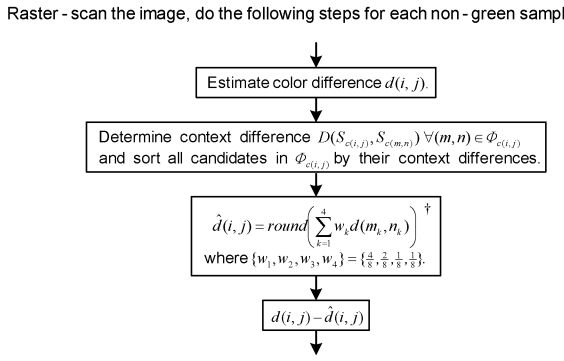

Fig. 13. How to handle the nongreen plane of a CFA image in CMBP.

known green samples as shown in Fig. 5. This is because the green channel has a double sampling rate as compared with the other channels in a CFA image and, hence, provides a more reliable context for matching.

In the proposed compression scheme, as green samples are encoded first in raster sequence, all green samples are known in the decoder, and, hence, the support of a nongreen sample can be noncausal while the support of a green sample has to be causal. This noncausal support tightly and completely encloses the sample of interest. It models image features such as intensity gradient, edge orientation, and textures better such that more accurate support matching can be achieved.

III. Adaptive Color Difference Estimation

When compressing the nongreen color plane, color difference information is exploited to remove the color spectral dependency. This section shows our proposed method for estimating the color difference value of a pixel without having a known green sample of the pixel.

Let $c(m,n)$ be the intensity value of the available color sample (either red or blue) at a nongreen sampling position (m,n). The green-red (green-blue) color difference of pixel (m,n), $d(m,n)$, is obtained by

$$d(m,n) = \hat{g}(m,n) - c(m,n) \tag{9}$$

where $\hat{g}(m,n)$ represents the estimated intensity value of the missing green component at position (m,n).

In the proposed estimation, $\hat{g}(m,n)$ is adaptively determined according to the horizontal gradient δH and the vertical gradient δV at (m,n) as follows:

$$\hat{g}(m,n) = \text{round}\left(\frac{\delta H \times G_V + \delta V \times G_H}{\delta H + \delta V}\right) \tag{10}$$

where $G_H = (g(m,n-1) + g(m,n+1))/(2)$ and $G_V = (g(m-1,n) + g(m+1,n))/(2)$ denote, respectively, the preliminary green estimates obtained by linearly interpolating the adjacent green samples horizontally and vertically. Note that, in (10), the missing green value is determined in such a way that a preliminary estimate contributes less if the gradient in the corresponding direction is larger. The weighing mechanism will automatically direct the estimation process along an edge if there is.

To simplify the estimation of $\hat{g}(m,n)$, one can check if pixel (m,n) is in a homogenous region by comparing the direction of (m,n)'s four neighboring green samples in $S_{c(m,n)}$. A straight forward estimation of $\hat{g}(m,n)$ can then be performed if it is. Specifically, we have (11), shown at the bottom of the page. In other words, as far as (10) is concerned, we have

$$\begin{cases} \delta H = 0 \\ \delta V = 1 \end{cases} \quad \text{if } \text{Dir}(a,b) = W, \forall (a,b) \in S_{c(m,n)}$$

and

$$\begin{cases} \delta H = 1 \\ \delta V = 0 \end{cases} \quad \text{if } \text{Dir}(a,b) = N, \forall (a,b) \in S_{c(m,n)}$$

when pixel (m,n) is in a homogenous region. Remind that the green plane is encoded first, and, hence, the directions of all green samples are available for the detection.

When pixel (m,n) is not in a homogenous region or the common direction of all green samples in $S_{c(m,n)}$ is not N or W, a more sophisticated approach is used to estimate gradients δH and δV for realizing (10). Specifically, they are determined by averaging all local green gradients in the same direction within a 5×5 window as

$$\delta H = \frac{1}{5} \sum_{\substack{(p,q) \in \{(m-1,n-2),\\ (m+1,n-2),(m,n-1),\\ (m-1,n),(m+1,n)\}}} |g(p,q) - g(p,q+2)|$$

and

$$\delta V = \frac{1}{5} \sum_{\substack{(p,q) \in \{(m-2,n-1),\\ (m-2,n+1),(m-1,n),\\ (m,n-1),(m,n+1)\}}} |g(p,q) - g(p+2,q)|. \tag{12}$$

To reduce the effort, a simpler approach can be used to estimate δH and δV with the four adjacent green samples in $S_{c(m,n)}$ as follows:

$$\delta H = |g(m,n-1) - g(m,n+1)|$$

and

$$\delta V = |g(m-1,n) - g(m+1,n)|. \tag{13}$$

In this paper, all simulation results related to the proposed algorithm were obtained with (12) instead of (13), unless it is specified otherwise.

$$\hat{g}(m,\mathrm{n}) = \begin{cases} \text{round}(G_H), & \text{if } \text{Dir}(a,b) = W, \forall (a,b) \in S_{c(m,n)} \\ \text{round}(G_V), & \text{if } \text{Dir}(a,b) = N, \forall (a,b) \in S_{c(m,n)} \end{cases} \tag{11}$$

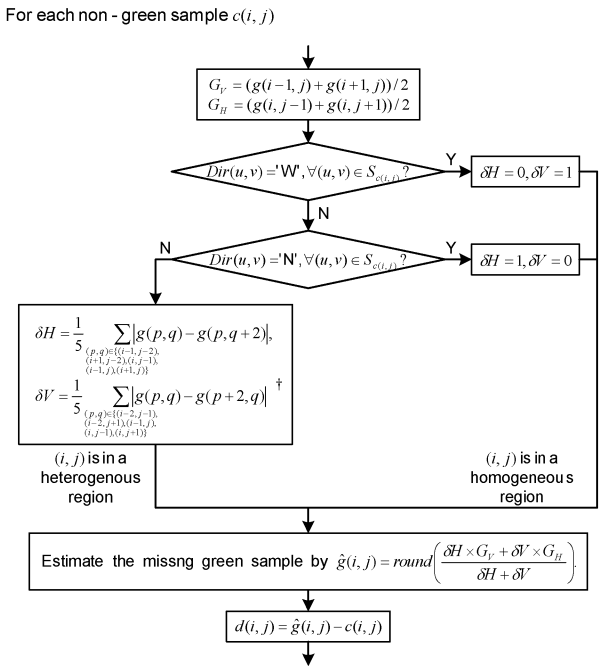

Fig. 14. How to estimate the color difference value of a nongreen sample.

Fig. 14 summaries how to estimate the color difference value of a nongreen sample in a CFA image. The proposed method works with CMBP as shown in Fig. 13 to produce a residue plane associated with the nongreen subimage. One can skip the classification of a region [i.e., the realization of (11)] by bypassing the two decision steps in the flowchart shown in Fig. 14. Fig. 15 shows the correlation of the residues obtained under the two different conditions. For testing image 8, the zero-order entropy value of $\{d(i,j) - \hat{d}(i,j) | (i,j) \in \text{ nongreen subimage}\}$ obtained with region classification and that obtained without region classification are, respectively, 5.434 and 5.437 bpp. The reduction in entropy may not be significant when region classification is exploited, but the reduction in complexity is considerable as the realization of (12) [or (13)] can be saved in this case.

IV. Proposed Compression Scheme

Fig. 3 shows the structure of the proposed compression scheme. In the encoding phase, a CFA image is first divided into a green subimage and a nongreen subimage. The green subimage is coded first and the nongreen subimage follows based on the green subimage as a reference.

To code a subimage, the subimage is raster-scanned and each pixel is predicted with its four neighboring pixels by using the prediction scheme proposed in Section II. The prediction error of pixel (i,j) in the CFA image, say $e(i,j)$, is given by (14), shown at the bottom of the page, where $g(i,j)$ and $d(i,j)$ are, respectively, the real green sample value and the color difference value of pixel (i,j). $d(i,j)$ is estimated by the method described in Section III. $\hat{g}(i,j)$ and $\hat{d}(i,j)$, respectively, represent the predicted green intensity value and the predicted color difference value of pixel (i,j). The error residue $e(i,j)$ is then mapped to a nonnegative integer as follows to reshape its value distribution to an exponential one from a Laplacian one.

$$E(i,j) = \begin{cases} -2e(i,j), & \text{if } e(i,j) \le 0 \\ 2e(i,j) - 1, & \text{otherwise.} \end{cases} \tag{15}$$

The $E(i,j)$'s from the green subimage are raster scanned and coded with Rice code first. The $E(i,j)$'s from the nongreen subimage are further decomposed into two residue subplanes. One carries the $E(i,j)$'s originated from the red CFA samples while the other one carries those originated from the blue CFA samples. The two residue subplanes are then raster scanned and coded with Rice code as well. Their order of processing does not matter as there is no interdependency among these two residue subplanes. That they are separately handled is just because the Rice code can be made adaptive to their statistical properties in such an arrangement. For reference, the residue subplanes originated from the red, the green and the blue CFA samples are, respectively, referred to as E_R, E_G, and E_B.

Rice code is employed to code $E(i,j)$ because of its simplicity and high efficiency in handling exponentially distributed sources. When Rice code is used, each mapped residue $E(i,j)$ is split into a quotient $Q = \text{floor}(E(i,j)/2^k)$ and a remainder $R = E(i,j) \text{mod}(2^k)$, where parameter k is a nonnegative integer. The quotient and the remainder are then saved for storage or transmission. The length of the codeword used for representing $E(i,j)$ is k-dependent and is given by

$$L(E(i,j)|k) = \text{floor}\left(\frac{E(i,j)}{2^k}\right) + 1 + k. \tag{16}$$

Parameter k is critical to the compression performance as it determines the code length of $E(i,j)$. For a geometric source $\boldsymbol{S}$ with distribution parameter $\rho \in (0,1)$ (i.e., $\text{Prob}(\boldsymbol{S} = s) = (1-\rho)\rho^s$ for $s = 0,1,2,\ldots$), the optimal coding parameter k is given as

$$k = \max\left\{0, \text{ceil}\left(\log_2\left(\frac{\log\phi}{\log\rho^{-1}}\right)\right)\right\} \tag{17}$$

where $\phi = (\sqrt{5}+1)/2$ is the golden ratio [24]. Since the expectation value of the source is given by $\mu = \rho(1-\rho)^{-1}$, as long as μ is known, parameter ρ, and, hence, the optimal coding parameter k for the whole source can be determined easily.

$$e(i,j) = \begin{cases} g(i,j) - \hat{g}(i,j), & \text{if pixel } (i,j) \text{ is in green subimage} \\ d(i,j) - \hat{d}(i,j), & \text{if pixel } (i,j) \text{ is in nongreen subimage} \end{cases} \tag{14}$$

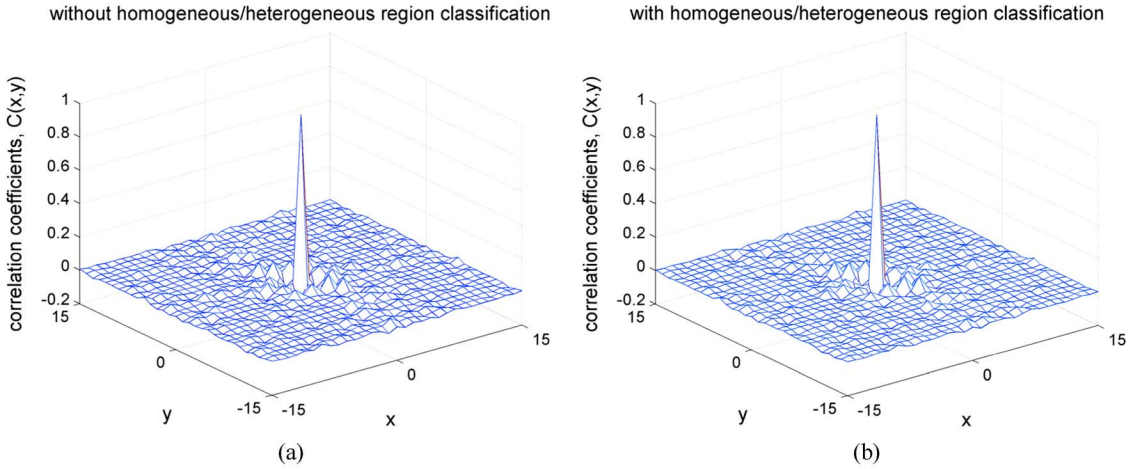

Fig. 15. Correlation among the prediction residues associated with the nongreen subimage of testing image 8 (a) without region classification and (b) with region classification in determining $d(i,j)$.

In the proposed compression scheme, μ is estimated adaptively in the course of encoding E_R, E_G, and E_B. In particular, it is estimated by

$$\tilde{\mu} = \text{round}\left(\frac{\alpha\tilde{\mu}_p + M_{i,j}}{1+\alpha}\right) \quad \text{and}$$
$$M_{i,j} = \left(\frac{1}{4}\sum_{(a,b)\in\zeta_{i,j}} E(a,b)\right) \tag{18}$$

where $\tilde{\mu}$ is the current estimate of μ for selecting the k to determine the codeword format of the current $E(i,j)$, $\tilde{\mu}_p$ is the previous estimate of $\tilde{\mu}$, $M_{i,j}$ is the local mean of $E(i,j)$ in a local region defined by set $\zeta_{i,j}$, and α is a weighting factor which specifies the significance of $\tilde{\mu}_p$ and $M_{i,j}$ when updating $\tilde{\mu}$. Set $\zeta_{i,j}$ is a set of four processed pixel locations which are closest to pixel (i,j) and, at the same time, possess samples of the same color as pixel (i,j) does. When coding E_G, it is defined to be $\{(i,j-2),(i-1,j-1),(i-2,j),(i-1,j+1)\}$. For coding E_R and E_B, set $\zeta_{i,j}$ is defined to be $\{(i,j-2),(i-2,j-2),(i-2,j),(i-2,j+2)\}$. $\tilde{\mu}$ is updated for each $E(i,j)$. The initial value of $\tilde{\mu}_p$ is 0 for all residue subplanes.

Experimental results showed that $\alpha = 1$ can provide a good compression performance. Fig. 16 shows how parameter α affects the final compression ratio of the proposed compression scheme. Curve R, G, and B, respectively, show the cases when coding E_R, E_G, and E_B. The curve marked with "All" shows the overall performance when all residue subplanes are compressed with a common α value.

The decoding process is just the reverse process of encoding. The green subimage is decoded first and then the nongreen subimage is decoded with the decoded green subimage as a reference. The original CFA image is then reconstructed by combining the two subimages.

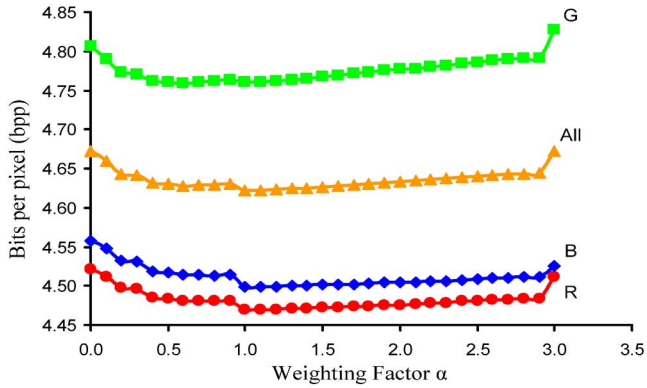

Fig. 16. Average output bit rates of the proposed compression scheme achieved with different α values.

V. Compression Performance

Simulations were carried out to evaluate the performance of the proposed compression scheme. Twenty-four 24-bit color images of size 512×768 each as shown in Fig. 8 were subsampled according to the Bayer pattern to form a set of 8-bit testing CFA images. They were then directly coded by the proposed compression scheme for evaluation. Some representative lossless compression schemes such as JPEG-LS [21], JPEG 2000 (lossless mode) [22] and LCMI [23] were also evaluated for comparison.

Table I lists the average output bit rates of the CFA images achieved by various compression schemes in terms of bits per pixel. It clearly shows that the proposed scheme outperforms all other evaluated schemes in all testing images. Especially for the images which contain many edges and fine textures such as images 1, 5, 8, 13, 20, and 24, the bit rates achieved by the proposed scheme are at least 0.34 bpp lower than the corresponding bit rates achieved by LCMI, the scheme offers the second best compression performance. These results demonstrate that the proposed compression scheme is robust to remove the CFA data dependency even though the image contains complicated structures. On average, the proposed scheme yields a bit rate as low as

TABLE I
ACHIEVED BIT RATES OF VARIOUS LOSSLESS COMPRESSION SCHEMES IN TERMS OF BITS PER PIXEL

Image	JPEG -LS	JPEG 2000	LCMI	Ours	Image	JPEG -LS	JPEG 2000	LCMI	Ours
1	6.403	5.816	5.824	5.478	**13**	6.747	6.372	6.503	6.138
2	6.787	5.134	4.629	4.331	**14**	6.289	5.555	5.487	5.170
3	5.881	4.216	3.965	3.746	**15**	6.317	4.656	4.396	4.102
4	6.682	4.931	4.606	4.379	**16**	5.289	4.552	4.521	4.376
5	6.470	5.947	5.859	5.409	**17**	4.965	4.547	4.499	4.286
6	5.871	5.210	5.139	4.881	**18**	6.184	5.570	5.538	5.284
7	5.974	4.500	4.299	3.960	**19**	5.470	4.909	4.898	4.711
8	6.295	5.899	5.966	5.570	**20**	4.317	4.026	4.054	3.541
9	5.074	4.391	4.319	4.188	**21**	5.467	5.039	4.983	4.803
10	5.395	4.556	4.415	4.227	**22**	6.188	5.218	5.060	4.847
11	5.370	4.986	4.952	4.683	**23**	6.828	4.525	3.960	3.847
12	5.628	4.485	4.307	4.089	**24**	5.719	5.223	5.257	4.873
					Avg.	**5.900**	**5.011**	**4.893**	**4.622**

TABLE II
AVERAGE BIT RATES (IN BITS/PIXEL) FOR CODING NONGREEN SUBIMAGES WITH THE PROPOSED COMPRESSION SCHEME WHEN USING A PARTICULAR ESTIMATION METHOD TO ESTIMATE A MISSING GREEN SAMPLE FOR REFERENCE

Image	The proposed compression scheme with					Image	The proposed compression scheme with				
	BI	ESI	ADI	Method proposed in Section 3			BI	ESI	ADI	Method proposed in Section 3	
				using (13)	using (12)					using (13)	using (12)
1	5.414	5.411	5.349	5.299	5.245	**13**	5.969	5.967	5.981	5.963	5.918
2	4.366	4.365	4.367	4.343	4.319	**14**	5.030	5.028	5.006	4.980	4.935
3	3.724	3.721	3.709	3.682	3.664	**15**	4.192	4.186	4.161	4.162	4.145
4	4.314	4.312	4.304	4.305	4.275	**16**	4.295	4.291	4.275	4.206	4.152
5	5.312	5.301	5.267	5.221	5.191	**17**	4.202	4.201	4.163	4.147	4.132
6	4.851	4.849	4.802	4.744	4.684	**18**	5.117	5.116	5.126	5.120	5.089
7	3.890	3.893	3.881	3.868	3.858	**19**	4.705	4.703	4.627	4.579	4.552
8	5.574	5.570	5.447	5.326	5.300	**20**	3.701	3.698	3.676	3.665	3.663
9	4.154	4.156	4.113	4.090	4.071	**21**	4.737	4.735	4.723	4.703	4.674
10	4.173	4.171	4.142	4.102	4.081	**22**	4.732	4.732	4.730	4.734	4.697
11	4.625	4.618	4.586	4.526	4.483	**23**	3.844	3.841	3.838	3.837	3.827
12	4.085	4.083	4.063	4.014	3.977	**24**	4.787	4.785	4.771	4.705	4.672
						Avg.	**4.575**	**4.572**	**4.546**	**4.513**	**4.484**

4.622 bpp. It is, respectively, around 78.3%, 92.1%, and 94.5% of those achieved by JPEG-LS, JPEG2000, and LCMI.

In the proposed compression scheme, the nongreen subimage is processed in the color difference domain. Accordingly, the missing green samples in the subimage have to be estimated for extracting the color difference information of the nongreen subimage. An estimation method for estimating the missing green samples and its simplified version [using (13) instead of (12) to estimate δH and δV], are proposed in Section III. Obviously, one can make use of some other estimation methods such as bilinear interpolation [9] (BI), edge sensing interpolation [8] (ESI) and adaptive directional interpolation [4] (ADI) to achieve the same objective.

For comparison, a simulation was carried out to evaluate the performance of these methods when they were used to compress a nongreen subimage with the proposed compression scheme. In this study, only the nongreen subimages are involved as the compression of green subimages does not involve the estimation of missing green components. In the realization of BI, a missing green sample is estimated by rounding the average value of its four surrounding known green samples. For ESI, the four surrounding known green samples are weighted before averaging. The weights are determined according to the gradients among the four known green samples [8]. ADI is a directional linear-based interpolation method in which the interpolation direction is determined by comparing the horizontal and vertical green gradients to a predefined threshold [4]. The threshold value was set to be 30 in our simulation as it provided the best compression result for the training set.

Table II reveals the average bit rates of the outputs achieved by the proposed compression scheme when different methods were used to estimate the missing green samples in the nongreen subimages. It shows that the adaptive estimation methods proposed in Section III are superior to the other evaluated estimation methods. On average, the best proposed estimation method achieves a bit rate of 4.484 bpp which is around 0.1 bpp lower than that achieved by BI.

While Table II reports the compression performance of the proposed compression scheme and its various variants, Table III lists their complexity cost paid for producing all prediction

TABLE III
AVERAGE COMPLEXITY (IN OPERATIONS/PIXEL) FOR DIFFERENT VARIANTS OF THE PROPOSED COMPRESSION SCHEME TO GENERATE PREDICTION RESIDUES OF BOTH GREEN AND NONGREEN SUBIMAGES

Methods used to estimate missing green samples in non-green planes	ADD	MUL	CMP	SHT	ABS	Total
BI	21.70	0.00	6.00	3.70	3.00	**34.4**
ESI	42.70	12.00	6.00	2.70	9.00	**72.4**
ADI	23.45	0.00	9.86	3.70	5.00	**42.0**
Method in Section 3 using (13)	22.35	2.65	7.93	3.70	3.00	**39.6**
Method in Section 3 using (12)	29.43	2.65	7.93	3.70	3.00	**46.7**

residues of both green and nongreen planes. It is measured in terms of the average number of operations required per pixel in our simulations. Operations including addition (ADD), multiplication (MUL), bit-shift (SHT), comparison (CMP), and taking absolute value (ABS) are all taken into account.

The proposed compression scheme is composed of four functional components. A study was carried out to evaluate the contribution of each component to the overall performance of the scheme. The same set of 24 testing CFA images were used again in the evaluation. In particular, when the prediction components are switched off [i.e., $\hat{g}(i,j) = \hat{d}(i,j) = 0$ in Fig. 3(a)], the zero-order entropy values of $\{e(i,j)|(i,j) \in \text{green subimage}\}$ and $\{e(i,j)|(i,j) \in \text{nongreen subimage}\}$ are, respectively, 7.114 and 6.295 bpp on average, which are around 40.3% and 34.2% higher than the case when the prediction components are on. As for the component of color difference estimation, the proposed adaptive color difference estimation scheme provided a nongreen residue plane of zero-order entropy 4.690 bpp on average, which is 0.114 bpp lower than that provided by using bilinear interpolation instead. To show the contribution of the proposed adaptive Rice code encoding scheme, we encoded $E(i,j)$ with the conventional Rice code instead of the proposed one for comparison. In its realization, the coding parameter k for coding a subimage is fixed and determined with (17). The parameter μ is estimated to be the mean of $E(i,j)$ in the subimage. After all, it achieved an average bit rate of 5.084 bpp, which is 0.462 bpp higher than that achieved by using the proposed adaptive Rice code encoding scheme.

When the proposed compression scheme (with (12)) was implemented in software with C++ programming language, the average execution time to compress a 512×768 CFA image on a 2.8-GHz Pentium 4 PC with 512-MB RAM is around 0.11 s.

VI. CONCLUSION

In this paper, a lossless compression scheme for Bayer images is proposed. This scheme separates a CFA image into a green subimage and a nongreen subimage and then encodes them separately with predictive coding. The prediction is carried out in the intensity domain for the green subimage while it is carried out in the color difference domain for the nongreen subimage. In both cases, a context matching technique is used to rank the neighboring pixels of a pixel for predicting the existing sample value of the pixel. The prediction residues originated from the red, the green, and the blue samples of the CFA images are then separately encoded.

The value distribution of the prediction residue can be modeled as an exponential distribution, and, hence, the Rice code is used to encode the residues. We assume the prediction residue is a local variable and estimate the mean of its value distribution adaptively. The divisor used to generate the Rice code is then adjusted accordingly so as to improve the efficiency of Rice code.

Experimental results show that the proposed compression scheme can efficiently and effectively decorrelate the data dependency in both spatial and color spectral domains. Consequently, it provides the best average compression ratio as compared with the latest lossless Bayer image compression schemes.

REFERENCES

[1] B. E. Bayer, *Color Imaging Array*. Rochester, NY: Eastman Kodak Company, 1976, U.S. 3 971 065.

[2] R. Lukac and K. N. Plataniotis, "Color filter arrays: Design and performance analysis," *IEEE Trans. Consum. Electron.*, vol. 51, no. 4, pp. 1260–1267, Apr. 2005.

[3] J. F. Hamilton and J. E. Adams, *Adaptive Color Plan Interpolation in Single Sensor Color Electronic Camera*. Rochester, NY: Eastman Kodak Company, 1997, U.S. 5 629 734.

[4] R. H. Hibbard, *Apparatus and Method for Adaptively Interpolating a Full Color Image Utilizing Luminance Gradients*. Rochester, NY: Eastman Kodak Company, 1995, U.S. 5 382 976.

[5] 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.

[6] X. L. Wu and N. Zhang, "Primary-consistent soft-decision color demosaicking for digital cameras (patent pending)," *IEEE Trans. Image Process.*, vol. 13, no. 9, pp. 1263–1274, Sep. 2004.

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

[8] R. Lukac and K. N. Plataniotis, "Data adaptive filters for demosaicking: A framework," *IEEE Trans. Consum. Electron.*, vol. 51, no. 2, pp. 560–570, May 2005.

[9] T. Sakamoto, C. Nakanishi, and T. Hase, "Software pixel interpolation for digital still cameras suitable for a 32-bit MCU," *IEEE Trans. Consum. Electron.*, vol. 44, no. 4, pp. 1342–1352, Nov. 1998.

[10] T. Toi and M. Ohta, "A subband coding technique for image compression in single CCD cameras with Bayer color filter arrays," *IEEE Trans. Consum. Electron.*, vol. 45, no. 1, pp. 176–180, Feb. 1999.

[11] X. Xie *et al.*, "A novel method of lossy image compression for digital image sensors with Bayer color filter arrays," in *Proc. IEEE Int. Symp. Circuits and Systems*, Kobe, Japan, 2005, pp. 4995–4998.

[12] S. Y. Lee and A. Ortega, "A novel approach of image compression in digital cameras with a Bayer color filter array," in *Proc. IEEE Int. Conf. Image Processing*, Thessaloniki, Greece, 2001, pp. 482–485.

[13] C. C. Koh, J. Mukherjee, and S. K. Mitra, "New efficient methods of image compression in digital cameras with color filter array," *IEEE Trans. Consum. Electron.*, vol. 49, no. 4, pp. 1448–1456, Nov. 2003.

[14] N. X. Lian *et al.*, "Reversing demosaicking and compression in color filter array image processing: Performance analysis and modeling," *IEEE Trans. Image Process.*, vol. 15, no. 11, pp. 3261–3278, Nov. 2006.

[15] Y. T. Tsai, "Color image compression for single-chip cameras," *IEEE Trans. Electron Devices*, vol. 38, no. 5, pp. 1226–1232, May 1991.
[16] A. Bruna *et al.*, "Predictive differential modulation for CFA compression," in *Proc. 6th Nordic Signal Processing Symp.*, 2004, pp. 101–104.
[17] S. Battiato *et al.*, "Coding techniques for CFA data images," in *Proc. Int. Conf. Image Analysis and Processing*, Mantova, Italy, 2003, pp. 418–423.
[18] A. Bazhyna, A. Gotchev, and K. Egiazarian, "Near-lossless compression algorithm for Bayer pattern color filter arrays," *Proc. SPIE*, vol. 5678, pp. 98–209, 2005.
[19] B. Parrein, M. Tarin, and P. Horain, "Demosaicking and JPEG2000 compression of microscopy images," in *Proc. IEEE Int. Conf. Image Processing*, Singapore, 2004, pp. 521–524.
[20] R. Lukac and K. N. Plataniotis, "Single-sensor camera image compression," *IEEE Trans. Consum. Electron.*, vol. 52, no. 2, pp. 299–307, 2006.
[21] *Information Technology—Lossless and Near-Lossless Compression of Continuous-Tone Still Images (JPEG-LS)*, ISO/IEC Standard 14495-1, 1999.
[22] *Information Technology—JPEG 2000 Image Coding System—Part 1: Core Coding System*, INCITS/ISO/IEC Standard 15444-1, 2000.
[23] N. Zhang and X. L. Wu, "Lossless compression of color mosaic images," *IEEE Trans. Image Process.*, vol. 15, no. 6, pp. 1379–1388, Jun. 2006.
[24] A. Said, "On the determination of optimal parameterized prefix codes for adaptive entropy coding," Tech. Rep. HPL-2006-74, HP Lab., Palo Alto, CA, 2006.

King-Hong Chung received the B.Eng. (Hons.) degree in electronic and information engineering from The Hong Kong Polytechnic University in 2001, where he is currently pursuing the Ph.D. degree.

His current research interests include digital image halftoning, image compression, and digital camera image processing.

Yuk-Hee Chan (M'92) received the B.Sc. degree with honors in electronics from the Chinese University of Hong Kong in 1987 and the Ph.D. degree in signal processing from The Hong Kong Polytechnic University in 1992.

Between 1987 and 1989, he was an R&D Engineer in the Elec & Eltek Group, Hong Kong. He joined The Hong Kong Polytechnic University in 1992 and he is now an Associate Professor in the Department of Electronic and Information Engineering. He has published over 110 research papers in various international journals and conferences. His research interests include image and video compression, image restoration, halftoning, demosaicking, and fast computational algorithms in digital signal processing.

Dr. Chan is a member of the IET. He was the Chairman of the IEEE Hong Kong Joint Chapter of CAS and COM in 2003–2004.