Lossless Compression of Color Mosaic Images

Ning Zhang and Xiaolin Wu, *Senior Member, IEEE*

***Abstract*—Lossless compression of color mosaic images poses a unique and interesting problem of spectral decorrelation of spatially interleaved R, G, B samples. We investigate reversible lossless spectral-spatial transforms that can remove statistical redundancies in both spectral and spatial domains and discover that a particular wavelet decomposition scheme, called Mallat wavelet packet transform, is ideally suited to the task of decorrelating color mosaic data. We also propose a low-complexity adaptive context-based Golomb–Rice coding technique to compress the coefficients of Mallat wavelet packet transform. The lossless compression performance of the proposed method on color mosaic images is apparently the best so far among the existing lossless image codecs.**

***Index Terms*—Context quantization, entropy coding, digital camera, image compression.**

I. Introduction

MOST digital cameras use image sensors that sample only one of the three primary colors at each pixel position. Specifically, each pixel is covered with a filter and records just one of the three primary colors: red, green or blue. These primary color samples are interleaved in a two-dimensional (2-D) grid, or color mosaic, resembling a three-color checkerboard. The most popular single CCD color mosaic pattern is the one proposed by Bayer [1]. To reconstruct the true continuous-tone color, a procedure called color demosaicking is needed to interpolate the other two missing primary colors at each pixel. The image quality of digital cameras largely depends on the performance of the color demosaicking process.

Image data compression is an important component of digital camera design and digital photography. It is more than just an issue of saving storage and bandwidth, but rather to be considered in light of overall system performance and functionality, particularly in relation to color demosaicking. Currently, all digital cameras carry out color demosaicking prior to compression, apparently due to the considerations of easy user interface and device compatibility. However, industrial policy and standard issues aside, in our opinion, this design is suboptimal. Color demosaicking triples the amount of raw data by generating R, G, B bands via color interpolation. Ironically, the task of compression needs to decorrelate the three bands, which essentially attempts to reverse engineer the color interpolation process of demosaicking. This demosaicking-first and compression-later design unnecessarily increases algorithm complexity, reduces compression ratio, and burdens the on-camera I/O bandwidth.

In this paper, we propose to compress and store the color mosaic data directly, and perform demosaicking to reconstruct the R, G, B bands afterward, possibly offline. This relieves the camera from the tasks of color demosaicking and color decorrelation and also reduces the amount of input data to compression codec in the first place. The new workflow can potentially reduce on-camera computing power and I/O bandwidth. More importantly, the new design allows lossless or near-lossless compression of raw mosaic data, which is the main theme of this paper.

For many high-end digital photography applications, such as digital archiving of precious museum arts and relics, professional advertising, and digital cinema for which high image quality is paramount, it is crucial to have the original color mosaic data in lossless format. Our recent results in color demosaicking research [2] indicate that superior image quality can be obtained by more sophisticated color demosaicking algorithms than those implemented on camera, provided that original mosaic data are available. Furthermore, other image/video applications, such as super-resolution imaging and motion analysis, should also benefit from lossless compression of color mosaic data, in which even subpixel precision is much desired.

Lossless compression of mosaic color images poses a unique and interesting problem of spectral decorrelation (or more generally statistical modeling) of spatially interleaved R, G, B samples. Because a color mosaic image consists of interlaced R, G, B samples, existing decorrelation techniques such as DPCM, DCT, and wavelets may not work effectively by treating a mosaic image as a grayscale one. In this paper, we examine a number of interband coding techniques for lossless coding of color mosaic images. Our focus is on reversible lossless spectral-spatial transforms that can remove statistical redundancies in both spectral and spatial domains. Interestingly, we discover that a unique wavelet decomposition scheme, called the Mallat packet transform, is ideally suited to the task of decorrelating color mosaic data.

The presentation is organized as follows. Section II presents and evaluates some schemes of coding mosaic images by de-interleaving R, G, B samples prior to compression. In Section III, we consider an alternative approach of compressing color mosaic images directly without de-interleaving the color bands. We study the strength and weakness of both DPCM and wavelet-based lossless coding methods in the above two different approaches. Section IV offers a wavelet analysis of mosaic images. The analysis leads to a new wavelet decomposition scheme that is well suited for lossless coding of Bayer pattern mosaic data directly without de-interleaving. This new wavelet decomposition, which resembles the SPACL mode of JPEG 2000 standard

Manuscript received September 29, 2004; revised April 18, 2005. This work was supported in part by the Natural Sciences and Engineering Research Council of Canada under the NSERC-DALSA Industrial Research Chair in Digital Cinema. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Giovanni Poggi.

The authors are with the Department of Electrical and Computer Engineering, McMaster University, Hamilton, ON L8K 4K1 Canada (e-mail: ningzhang@mail.ece.mcmaster.ca; xwu@ece.mcmaster.ca).

has the nice property of decorrelating color samples both spatially and spectrally. Section V introduces a fast context-based Golomb–Rice coding scheme to compress the coefficients of the proposed wavelet transform. Section VI presents experimental results and Section VII concludes.

II. Deinterleaved Compression

Since most digital cameras use CCD sensor arrays of Bayer pattern, we are concerned with the lossless compression of color mosaic images of Bayer pattern, but the techniques to be developed in this paper can be generalized to other mosaic color sampling schemes. The Bayer color filter array and a resulting mosaic image are presented in Fig. 1.

Let $I_{(x,y)}$ be the color sample at pixel position (x, y), then the Bayer color mosaic pattern is defined by

$$I_{(x,y)} = \begin{cases} G_{(x,y)} & x-y \text{ is even} \\ R_{(x,y)} & x \text{ is odd, } y \text{ is even} \\ B_{(x,y)} & x \text{ is even, } y \text{ is odd.} \end{cases} \quad (1)$$

A natural way of compressing color mosaic images is to first deinterleave the three color channels, and then code each of the three down-sampled color channels individually. Specifically, the Bayer pattern

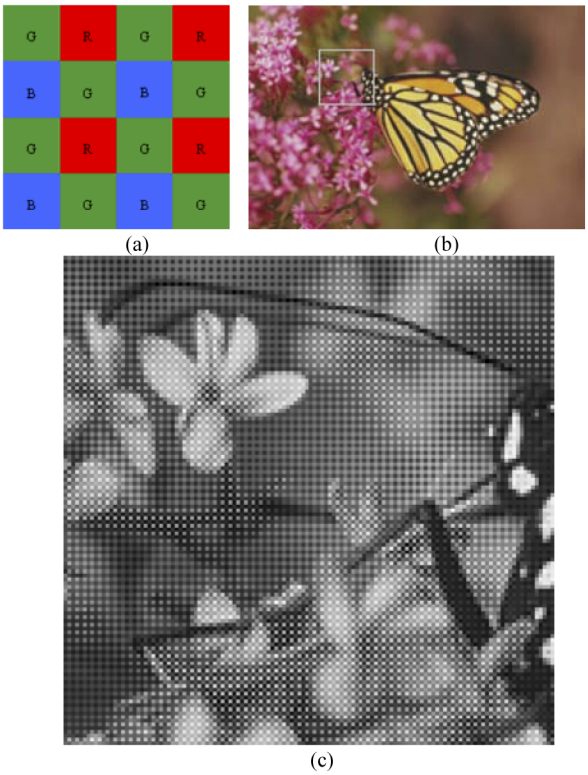
$$\begin{array}{cccccccc} G_{0,0} & R_{0,1} & G_{0,2} & R_{0,3} & G_{0,4} & R_{0,5} & G_{0,6} & R_{0,7} \\ B_{1,0} & G_{1,1} & B_{1,2} & G_{1,3} & B_{1,4} & G_{1,5} & B_{1,6} & G_{1,7} \\ G_{2,0} & R_{2,1} & G_{2,2} & R_{2,3} & G_{2,4} & R_{2,5} & G_{2,6} & R_{2,7} \\ B_{3,0} & G_{3,1} & B_{3,2} & G_{3,3} & B_{3,4} & G_{3,5} & B_{3,6} & G_{3,7} \\ G_{4,0} & R_{4,1} & G_{4,2} & R_{4,3} & G_{4,4} & R_{4,5} & G_{4,6} & R_{4,7} \\ B_{5,0} & G_{5,1} & B_{5,2} & G_{5,3} & B_{5,4} & G_{5,5} & B_{5,6} & G_{5,7} \\ G_{6,0} & R_{6,1} & G_{6,2} & R_{6,3} & G_{6,4} & R_{6,5} & G_{6,6} & R_{6,7} \\ B_{7,0} & G_{7,1} & B_{7,2} & G_{7,3} & B_{7,4} & G_{7,5} & B_{7,6} & G_{7,7} \end{array}$$

can be de-interleaved into the following three down-sampled color channels:

$$\begin{array}{cccccccc} G_{0,0} & & G_{0,2} & & G_{0,4} & & G_{0,6} & \\ & G_{1,1} & & G_{1,3} & & G_{1,5} & & G_{1,7} \\ G_{2,0} & & G_{2,2} & & G_{2,4} & & G_{2,6} & \\ & G_{3,1} & & G_{3,3} & & G_{3,5} & & G_{3,7} \\ G_{4,0} & & G_{4,2} & & G_{4,4} & & G_{4,6} & \\ & G_{5,1} & & G_{5,3} & & G_{5,5} & & G_{5,7} \\ G_{6,0} & & G_{6,2} & & G_{6,4} & & G_{6,6} & \\ & G_{7,1} & & G_{7,3} & & G_{7,5} & & G_{7,7} \end{array}$$

$$\begin{array}{cccc} R_{0,1} & R_{0,3} & R_{0,5} & R_{0,7} \\ R_{2,1} & R_{2,3} & R_{2,5} & R_{2,7} \\ R_{4,1} & R_{4,3} & R_{4,5} & R_{4,7} \\ R_{6,1} & R_{6,3} & R_{6,5} & R_{6,7} \end{array} \qquad \begin{array}{cccc} B_{1,0} & B_{1,2} & B_{1,4} & B_{1,6} \\ B_{3,0} & B_{3,2} & B_{3,4} & B_{3,6} \\ B_{5,0} & B_{5,2} & B_{5,4} & B_{5,6} \\ B_{7,0} & B_{7,2} & B_{7,4} & B_{7,6}. \end{array}$$

Let us develop a general framework for de-interleaved compression of mosaic images. First, we code the green channel before the other two channels, because the green channel has twice as many samples and, hence, higher intrachannel correlation. Once the green samples are coded, we utilize the interchannel correlation to compress red and blue channels, but one problem needs to be addressed. In the Bayer pattern, the green channel



(a) (b) (c)

Fig. 1. Bayer pattern and an example of mosaic image: (a) Bayer color filter array; (b) an original color image: Monarch; (c) Bayer pattern mosaic image of the selected region in (b). (Color version available online at http://ieeexplore.ieee.org.)

square sample grid. We need to transform the green quincunx array to rectangular array in preparing it for compression. There are many ways of transforming or deinterleaving the diamond sample grid into a square sample grid. We examine the following four.

1) **Merge:** The quincunx array is converted to rectangular array by shifting all odd columns one pixel to the left and form

$$\begin{array}{cccc} G_{0,0} & G_{0,2} & G_{0,4} & G_{0,6} \\ G_{1,1} & G_{1,3} & G_{1,5} & G_{1,7} \\ G_{2,0} & G_{2,2} & G_{2,4} & G_{2,6} \\ G_{3,1} & G_{3,3} & G_{3,5} & G_{3,7} \\ G_{4,0} & G_{4,2} & G_{4,4} & G_{4,6} \\ G_{5,1} & G_{5,3} & G_{5,5} & G_{5,7} \\ G_{6,0} & G_{6,2} & G_{6,4} & G_{6,6} \\ G_{7,1} & G_{7,3} & G_{7,5} & G_{7,7}. \end{array}$$

2) **Reversible de-interlacer:**

$$\begin{array}{cccc} G_{0,0} & G_{0,2} & G_{0,4} & G_{0,6} \\ g_{1,1} & g_{1,3} & g_{1,5} & g_{1,7} \\ G_{2,0} & G_{2,2} & G_{2,4} & G_{2,6} \\ g_{3,1} & g_{3,3} & g_{3,5} & g_{3,7} \\ G_{4,0} & G_{4,2} & G_{4,4} & G_{4,6} \\ g_{5,1} & g_{5,3} & g_{5,5} & g_{5,7} \\ G_{6,0} & G_{6,2} & G_{6,4} & G_{6,6} \\ g_{7,1} & g_{7,3} & g_{7,5} & g_{7,7}. \end{array}$$

passed through a vertical low-pass filter before merging with even columns, namely

$$g_{(x,y)} = \frac{1}{4}\left[(G_{(x-1,y-1)} + 2G_{(x,y)} + G_{(x-1,y+1)}\right]. \quad (2)$$

Since $g_{(x,y)}$ increases the dynamic range of G by one bit, the above transform becomes inefficient for lossless coding. In other words, if the binary representation of $G_{(x,y)}$ has N bits, $N+1$ bits are required to represent $g_{(x,y)}$ in order to have lossless inverse transform. Although $g_{(x,y)}$ and $G_{(x,y)}$ have the same dynamic range in (2), an extra bit is needed to resolve the parity of the sum for lossless reconstruction.

3) **Separation**

$$\begin{array}{cccc|cccc} G_{0,0} & G_{0,2} & G_{0,4} & G_{0,6} & G_{1,1} & G_{1,3} & G_{1,5} & G_{1,7} \\ G_{2,0} & G_{2,2} & G_{2,4} & G_{2,6} & G_{3,1} & G_{3,3} & G_{3,5} & G_{3,7} \\ G_{4,0} & G_{4,2} & G_{4,4} & G_{4,6} & G_{5,1} & G_{5,3} & G_{5,5} & G_{5,7} \\ G_{6,0} & G_{6,2} & G_{6,4} & G_{6,6} & G_{7,1} & G_{7,3} & G_{7,5} & G_{7,7}. \end{array}$$

4) Rotation

$$\begin{array}{ccccccc} & & & G_{0,0} & & & \\ & & G_{2,0} & G_{1,1} & G_{0,2} & & \\ & G_{4,0} & G_{3,1} & G_{2,2} & G_{1,3} & G_{0,4} & \\ G_{6,0} & G_{5,1} & G_{4,2} & G_{3,3} & G_{2,4} & G_{1,5} & G_{0,6} \\ G_{7,1} & G_{6,2} & G_{5,3} & G_{4,4} & G_{3,5} & G_{2,6} & G_{1,7} \\ & G_{7,3} & G_{6,4} & G_{5,5} & G_{4,6} & G_{3,7} & \\ & & G_{7,5} & G_{6,6} & G_{5,7} & & \\ & & & G_{7,7}. & & & \end{array}$$

After one of the above deinterleaving transforms, the green channel can be coded using any of the existing lossless image codecs, such as JPEG-LS and JPEG 2000 lossless mode. Table I lists the bit rates of the lossless image compression standards JPEG-LS [5] and JPEG-2000 [6] (lossless mode) on the outputs of three of the above deinterleaving transforms. In Table I, for each test image, the number in bold face represents the best result among all deinterleaving transforms. There is no single winning transform for all the images. Not surprisingly, the separation transform performs the worst on average because it disregards the correlation between the two resulting subimages of green samples. The compression results of the merge and rotation transforms are very close for a given lossless image codec. In our comparison study JPEG-LS achieves better lossless compression than JPEG-2000 on all test images for the de-interleaving methods of separation and merge. For the rotation method, we present only the results of JPEG-LS not those of JPEG 2000, because it is relatively easy to modify JPEG-LS to code the rotated image but very difficult to do the same with JPEG 2000.

Once the green channel is coded and made known to the decoder, it can be used as an anchor to facilitate the compression of red and blue channels by exploiting the spectral correlation. To this end, we estimate the missing green values from the existing green samples at the pixel positions where either red or blue sample is taken. We denote such estimates by $g_{(x,y)}$, where

TABLE I
LOSSLESS BIT RATES OF GREEN CHANNEL UNDER DIFFERENT DEINTERLEAVING TRANSFORMS WHILE BEING COMPRESSED BY JPEG-LS AND JPEG 2000

	Separation		Merge		Rotation
Image	J2K	JPEG-LS	J2K	JPEG-LS	JPEG-LS
Woman	5.251	5.207	5.167	5.086	**5.004**
Bike	5.374	5.029	5.237	4.944	**4.919**
Monarch	4.699	4.518	4.496	4.318	**4.166**
Wall	6.197	**5.943**	6.147	5.985	6.068
Boat	5.364	5.176	5.200	**5.031**	5.131
Windows	6.418	**6.060**	6.389	6.320	6.285
Landscape	6.815	6.585	6.635	**6.401**	6.434
Fence	5.201	5.078	5.094	5.075	**5.054**
Lighthouse	5.234	5.05	5.063	**4.896**	4.976
Average	5.617	5.405	5.492	5.340	**5.337**

TABLE II
LOSSLESS BIT RATES OF RED AND BLUE MOSAIC SAMPLES USING JPEG-LS

	Intra-channel		Bilinear Interpolation		CubicSpline Interpolation	
Image	R	B	R-g	B-g	R-g	B-g
Woman	5.185	5.273	4.751	4.792	**4.698**	**4.653**
Bike	4.812	5.254	**4.571**	5.003	4.647	**4.954**
Monarch	4.528	4.625	**4.323**	**4.334**	4.397	4.391
Wall	5.962	5.937	5.566	5.526	**5.421**	**5.343**
Boat	5.199	5.349	4.833	5.023	**4.726**	**4.912**
Windows	6.062	6.04	5.688	5.664	**5.505**	**5.501**
Landscape	6.594	6.531	6.085	6.138	**6.039**	**6.122**
Fence	5.118	5.051	4.713	4.774	**4.645**	**4.773**
Lighthouse	5.053	5.154	4.778	4.881	**4.752**	**4.854**
Average	5.390	5.468	5.034	5.126	**4.981**	**5.056**

than code $R_{(x,y)}$ and $B_{(x,y)}$ separately, we losslessly code the two color difference images

$$\alpha_{(x,y)} = R_{(x,y)} - g_{(x,y)} \quad \text{and} \quad \beta_{(x,y)} = B_{(x,y)} - g_{(x,y)}. \quad (3)$$

Since the decoder can make the same estimates $g_{(x,y)}$ as the encoder, it can reconstruct the original $R_{(x,y)}$ and $B_{(x,y)}$ from $\alpha_{(x,y)}$ and $\beta_{(x,y)}$ without any loss. The difference images $\alpha_{(x,y)}$ and $\beta_{(x,y)}$, which can be regarded approximately as two chrominance components, are more compressible than $R_{(x,y)}$ or $B_{(x,y)}$ because they are typically low-pass signals due to the interchannel correlation. Furthermore, the color difference images $\alpha_{(x,y)}$ and $\beta_{(x,y)}$ provide vital information in many color demosaicing algorithms [1], [7]–[9]. Therefore, lossless coding of $\alpha_{(x,y)}$ and $\beta_{(x,y)}$ serves dual purposes of lossless compression of color mosaic data and color demosaicking.

In estimating the missing green samples $g_{(x,y)}$, we have evaluated various interpolation schemes, including bilinear interpolation, bi-cubic B-Spline [9] and some nonlinear methods [7], [8]. Table II lists the lossless bit rates of the red and blue mosaic samples obtained by three coding schemes: 1) intrachannel coding of red and blue; 2) color difference coding with bilinear green interpolation; 3) color difference coding with bi-cubic B-spline green interpolation. It is clear from Table II that coding color differences is more effective than coding the red and blue channels individually. The coding gain is more than 7.5% on average, which is a significant margin by the standard of lossless image coding. The precision of green interpolation can improve the lossless compression of color mosaic images, but only marginally. Simple bilinear interpolation works satisfactorily.

Table III presents the overall lossless bit rates of JPEG-LS and JPEG 2000 lossless mode on the de-interleaved green channel

TABLE III
LOSSLESS BIT RATES OF DEINTERLEAVED MOSAIC IMAGES BY JPEG-LS AND JPEG 2000

	No interpolation		Bilinear interpolation	
Image	JPEG-LS	JPEG-2K	JPEG-LS	JPEG-2K
Woman	5.158	5.220	4.929	4.968
Bike	4.989	5.289	4.866	5.064
Monarch	4.447	4.613	4.323	4.448
Wall	5.967	6.152	5.766	5.855
Boat	5.153	5.323	4.980	5.085
Windows	6.186	6.388	5.998	6.086
Landscape	6.482	6.697	6.256	6.406
Fence	5.080	5.154	4.909	4.926
Lighthouse	5.000	5.161	4.863	4.960
Average	**5.384**	**5.555**	**5.210**	**5.311**

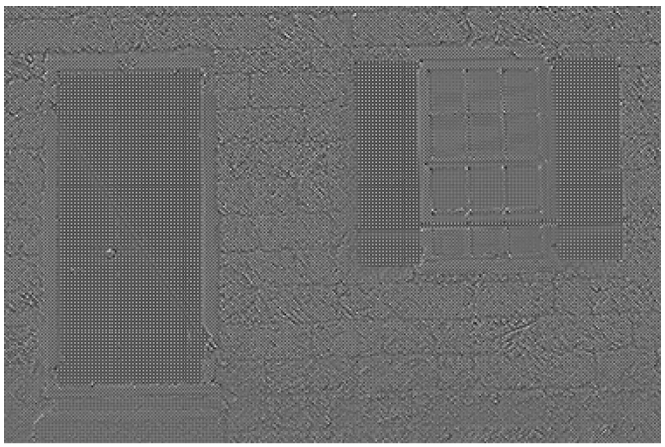
interpolation. For comparison purposes, we also give the results of coding red and blue channels directly without interband decorrelation.

III. INTERLEAVED COMPRESSION

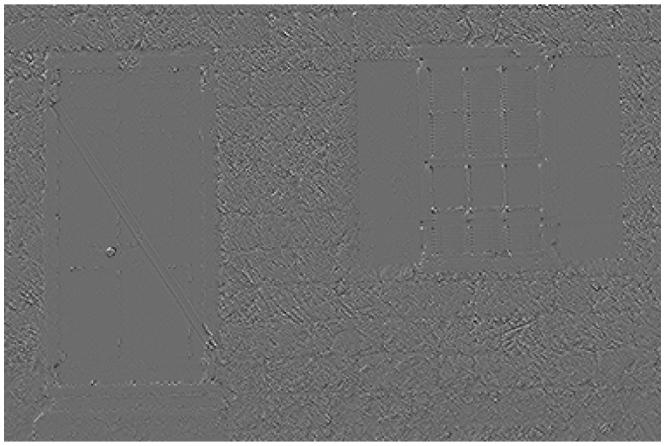
An alternative approach to lossless compression of color mosaic images is to process the mosaic data directly without de-interleaving the color channels. In other words, the compression algorithm pretends that the color mosaic image is a single-channel grrayscale image. This treatment has the advantage of simpler codec design and lower complexity than compression after de-interleaving. The simplest way is to apply a lossless image coding algorithm directly to raw color mosaic images without any preprocessing. For a quick assessment of different compression methods, when applied to mosaic images directly, the reader is referred to Table VI of Section VI (not placed here to save space) for the lossless bit rates of JPEG-LS and JPEG 2000 (using the 5-3 integer filter) standards on some common test images.

Interestingly, and somewhat surprisingly, JPEG-2000 outperforms JPEG-LS by a significant margin (more than 10%), when both applied to compress color mosaic images without de-interleaving. Recall from the proceeding section that the performance comparison between the two algorithms in the case of de-interleaved compression gave exactly opposite results. This reversal in relative coding efficiency is largely due to a fundamental difference in decorrelation mechanisms of the two algorithms: DPCM for JPEG-LS and lifting integer wavelet for JPEG 2000.

The DPCM scheme is suited to remove long term memory of a smooth signal in the spatial domain. It becomes ineffective on decorrelating mosaic images of periodic patterns. The energy of a mosaic image can be packed into the spatial-frequency domain of the wavelet far more efficiently than in the spatial domain. To expose this weakness of DPCM on mosaic images, in Fig. 2 we present the prediction residual images of JPEG-LS when being applied to a natural image [given in Fig. 6(a)] and its mosaic counterpart [given in Fig. 6(b)]. The DPCM residual signal of the mosaic image has significantly greater amplitude than that of the natural image. Moreover, the DPCM residuals still exhibit the original mosaic structure, with their statistics far from being



(a)



(b)

Fig. 2. Residual images of the median predictor of JPEG-LS (the mid-gray represents zero). (a) Residual image of Bayer mosaic image. (b) Residual image of the original green channel.

Let us compare in Fig. 3 the histograms of the prediction residual images Fig. 2(a) and Fig. 2(b) of JPEG-LS, for the mosaic image and the corresponding normal image, respectively. It is well known that the DPCM residuals of a normal image signal obey a Laplacian distribution, as being evident in Fig. 3(b), but this is no longer true for the DPCM residuals of a mosaic image. Note that the distribution of Fig. 3(a) is multimodal and asymmetric against the origin. The residuals of JPEG-LS for mosaic images deviate drastically from a Laplacian distribution, and they cannot even be modeled by a generalized Gaussian distribution. Unfortunately, the entropy code (Golomb–Rice code) of JPEG-LS assumes a Laplacian distribution of the prediction residuals. This severe mismatch between the model and the source also explains the poor performance of JPEG-LS on mosaic images. The problem will be corrected in the next two sections.

IV. WAVELET ANALYSIS OF MOSAIC IMAGES

In a sharp contrast to DPCM, the wavelet, being a tool of frequency–time analysis, can compactly characterize periodic

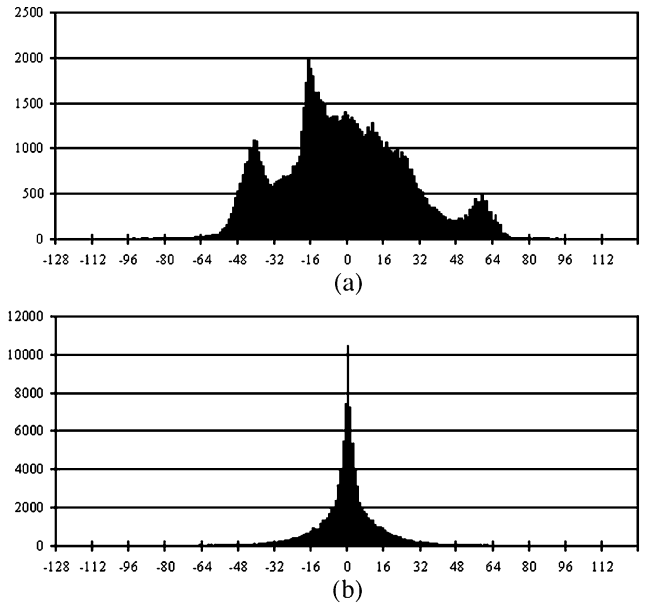


Fig. 3. Histograms of JPEG-LS residuals for mosaic image and the original green channel. (a) Histogram of the residual image in Fig. 2(a). (b) Histogram of the residual image in Fig. 2(b).

so-called Mallat packet wavelet transform for direct compression of mosaic images without de-interleaving. As we will see, the proposed wavelet transform simultaneously performs spatial and spectral decorrelation of color samples with a great ease and at a low cost.

Let us start by examining an interesting interplay between the Bayer pattern and the 2-D integer wavelet transform via separable one-dimensional (1-D) lifting. One level of the wavelet transform produces four subbands that have clear interpretations of the attributes of the Bayer color signal. As shown in Fig. 4, a 2-D wavelet transform (after decimation) and the Bayer pattern both have a 2×2 periodical sampling pattern. This correspondence makes 2-D wavelet transforms very efficient to represent the Bayer pattern in frequency–space domain. The effect of performing a 2-D wavelet transform on a mosaic image is illustrated by Fig. 5. In this example, the Bayer mosaic data of a uniform color image are transformed into four constant subbands, although the input mosaic image is, itself, a high-frequency signal.

We can explain the effect of Fig. 5 analytically using, for example, the 5-3 integer wavelet. Other wavelets, such as 9/7M, 5/11-C [10], behave similarly. The low- and high-pass filers of the 5-3 integer wavelet are

$$\mathbf{f}_L = \left(-\frac{1}{8}, \frac{1}{4}, \frac{3}{4}, \frac{1}{4}, -\frac{1}{8}\right) \quad \mathbf{f}_H = \left(-\frac{1}{2}, 1, -\frac{1}{2}\right). \tag{4}$$

After applying the 2-D low-pass filter $\mathbf{F}_{LL} = \mathbf{f}_L^T \mathbf{f}_L$ to the Bayer mosaic image, the LL subband can be interpreted as the luminance channel of the original full color image. In a window of smooth color, where red, green, and blue color components are approximately constants (i.e., $R_{(x,y)} \cong R$, $G_{(x,y)} \cong G$, $B_{(x,y)} \cong B$), the coefficients in the LL subband are

$$\begin{array}{ccccc} G & R & G & R \\ B & G & B & G \\ G & R & G & R \\ B & G & B & G \end{array} \qquad \begin{array}{cccc} LL & HL & LL & HL \\ LH & HH & LH & HH \\ LL & HL & LL & HL \\ LH & HH & LH & HH \end{array}$$

(a) (b)

Fig. 4. The 2×2 periodical sampling pattern of (a) Bayer mosaic data and (b) 2-D wavelet transform.

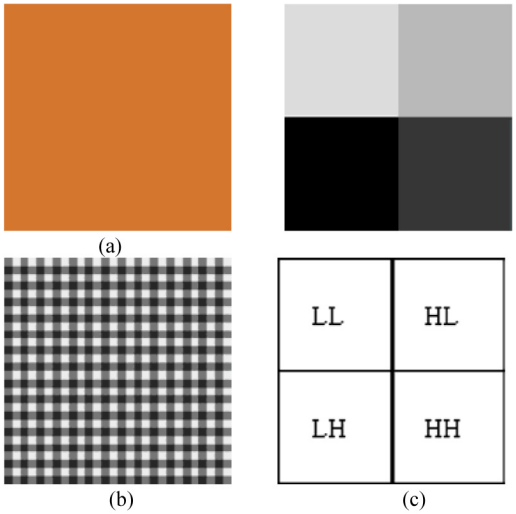


Fig. 5. Efficiency of representing Bayer pattern mosaic image in wavelet domain: (a) a uniform color image; (b) Bayer mosaic image; (c) 2-D wavelet coefficients. (Color version available online at http://ieeexplore.ieee.org.)

This linear combination of R, G, and B of rational coefficients is an approximation of the Y (luminance) component of the NTSC YUV color space. Interestingly, it is also exactly the same as the luminance component of reversible color transform adopted by JPEG 2000 in its lossless mode.

To understand the physical meanings of the three high subbands (HH, HL, and LH) of a mosaic image, we use a model of mosaic images that was originally developed for the purpose of color demosaicking [2]. In this model, a Bayer mosaic image [see Fig. 6(b)] is viewed as a sum of two component images

$$\begin{array}{ccccc} G_{0,0} & R_{0,1} & G_{0,2} & R_{0,3} & G_{0,4} \\ B_{1,0} & G_{1,1} & B_{1,2} & G_{1,3} & B_{1,4} \\ G_{2,0} & R_{2,1} & G_{2,2} & R_{2,3} & G_{2,4} \\ B_{3,0} & G_{3,1} & B_{3,2} & G_{3,3} & B_{3,4} \\ G_{4,0} & R_{4,1} & G_{4,2} & R_{4,3} & G_{4,4} \end{array}$$

$$= \begin{array}{ccccc} G_{0,0} & G_{0,1} & G_{0,2} & G_{0,3} & G_{0,4} \\ G_{1,0} & G_{1,1} & G_{1,2} & G_{1,3} & G_{1,4} \\ G_{2,0} & G_{2,1} & G_{2,2} & G_{2,3} & G_{2,4} \\ G_{3,0} & G_{3,1} & G_{3,2} & G_{3,3} & G_{3,4} \\ G_{4,0} & G_{4,1} & G_{4,2} & G_{4,3} & G_{4,4} \end{array}$$

$$+ \begin{array}{ccccc} 0 & \alpha_{0,1} & 0 & \alpha_{0,3} & 0 \\ \beta_{1,0} & 0 & \beta_{1,2} & 0 & \beta_{1,4} \\ 0 & \alpha_{2,1} & 0 & \alpha_{2,3} & 0 \\ \beta_{3,0} & 0 & \beta_{3,2} & 0 & \beta_{3,4} \\ 0 & \alpha_{4,1} & 0 & \alpha_{4,3} & 0. \end{array}$$

DOCKET ALARM

Explore Litigation Insights

Docket Alarm provides insights to develop a more informed litigation strategy and the peace of mind of knowing you're on top of things.

Real-Time Litigation Alerts

Keep your litigation team up-to-date with **real-time alerts** and advanced team management tools built for the enterprise, all while greatly reducing PACER spend.

Our comprehensive service means we can handle Federal, State, and Administrative courts across the country.

Advanced Docket Research

With over 230 million records, Docket Alarm's cloud-native docket research platform finds what other services can't. Coverage includes Federal, State, plus PTAB, TTAB, ITC and NLRB decisions, all in one place.

Identify arguments that have been successful in the past with full text, pinpoint searching. Link to case law cited within any court document via Fastcase.

Analytics At Your Fingertips

Learn what happened the last time a particular judge, opposing counsel or company faced cases similar to yours.

Advanced out-of-the-box PTAB and TTAB analytics are always at your fingertips.

API

Docket Alarm offers a powerful API (application programming interface) to developers that want to integrate case filings into their apps.

LAW FIRMS
Build custom dashboards for your attorneys and clients with live data direct from the court.

Automate many repetitive legal tasks like conflict checks, document management, and marketing.

FINANCIAL INSTITUTIONS
Litigation and bankruptcy checks for companies and debtors.

E-DISCOVERY AND LEGAL VENDORS
Sync your system to PACER to automate legal marketing.