Inter-Color Context Classifier for High Performance Lossless Bayer Image Compression

D.A. Mitchell
Rehov Brosh, Mazkeret Batya

H.B. Mitchell
Rehov Brosh, Mazkeret Batya

ABSTRACT

Switched prediction algorithms are widely used for lossless image compression including Bayer image compression. All switched predictions algorithms have the same structure consisting of two separate functions working in tandem: A local pixel pattern function, or context classifier, and a set of pixel-value prediction functions. For each local context a different prediction function is selected. In this article we describe a new switched prediction algorithm specifically for lossless Bayer image compression. The new algorithm uses generic context classifier which may be used with any set of prediction functions. We show that using the generic context classifier we obtain a substantial improvement in lossless Bayer image compression. The new context classifier is both simple and fast to implement with a low memory requirement.

General Terms

Lossless Image Compression, Bayer Image

Keywords

Lossless image compression, Bayer image compression, switched prediction, inter-color context, JPEG-LS, CALIC

1. INTRODUCTION

Switched prediction algorithms are an important class of algorithms used for lossless image compression. The algorithms are simple to implement, fast to run with a low memory requirement and are widely used in applications where complexity, speed and memory use are severely limited. Among the different switched prediction algorithms is the state-of-the-art JPEG-LS standard [1] which was developed by the Joint Photographic Experts Group (JPEG) for lossless compression of continuous-tone still images. Switched prediction algorithms may also be used for lossless compression of Bayer images. However, in this case, the compression performance is low [2, 3] unless the Bayer image is suitably preprocessed.

All switched predictions algorithms, including JPEG-LS, have the same structure consisting of two separate functions which work in tandem:

—A local pixel pattern function, or context classifier; and

—A set of pixel-value prediction functions.

The switched prediction algorithm works as follows: The input image is scanned from top-to-bottom and in each row, from left-to-right. At each pixel, the context classifier identifies the local pixel pattern. This then determines which function is to be used for predicting the pixel value. Then, instead of transmitting the pixel value, the error, or difference between the predicted pixel-value and the true pixel value is transmitted.

Although switched prediction algorithms all have the same structure they use a wide variety of different context classifiers and prediction functions [1, 3, 4, 5, 6, 7]. In this article we describe a new switched prediction algorithm designed specifically for lossless Bayer image compression. The new algorithm uses a new *generic* context classifier which may then be used with any set of prediction functions. The new classifier is both simple to implement and fast to run with a small memory requirement. Experimentally, we illustrate the new generic classifier on three different sets of prediction functions. We found the highest overall lossless compression was obtained using a set of prediction functions proposed by Jiang [6, 7].

The article is organized as follows. In Sect. 2 we describe the structure of a Bayer image and briefly review the different approaches used to compress it without losses. In Sect. 3 we present a detailed description of the switched lossless prediction algorithms. In Sect. 4 we describe the new context classifier and in Sect. 5 we present the experimental results obtained with the new classifier. Finally the article ends with the conclusion in Sect. 6

2. LOSSLESS BAYER IMAGE COMPRESSION

Most digital cameras use a single image sensor to capture the scene image. In these cameras, a color filter array (CFA) is placed before the image sensor which then records only one color at each pixel location. The most common CFA design is the Bayer pattern which consists of two green components (G_0 and G_1), one red component (R) and one blue component (B). To obtain the full color (RGB) image, a *demosaic* operation is performed which interpolates the missing color components [8, 9, 10].

For long-term storage and efficient transmission of a Bayer image, data compression is critical. Over the past decade, both lossy and lossless compression schemes have been used. Lossy compression schemes compress the Bayer image by discarding visually redundant information. This means that with a lossy compression algorithm only an approximation of the original image can be reconstructed. For critical applications e.g. archival storage, high-end photography or medical diagnosis [11, 12, 13, 14, 15], the original image must, however, be faithfully reconstructed. Thus in these, and similar, applications only lossless compression schemes, which preserve all the information of the original image, can be used.

In order to losslessly compress the Bayer image a simple procedure would be to form the full RGB image [8, 9, 10] and then apply a conventional lossless compression algorithm. However such a procedure is computationally intensive. Thus in recent years the approach adopted has been to compress the raw Bayer image without first demosaicing [16, 17, 18, 19].

In lossless image compression, the techniques used for removing redundant information may be broadly divided into three:

—**Spatial**. In this technique we exploit the spatial correlation between the image pixels. An example of such a technique is prediction error coding. In this method the image is raster scanned. At each pixel we estimate its value using previously scanned pixels and the prediction error is recorded.

—**Spectral**. In this technique we exploit the correlation between the different color channels. An example of such a technique is to predict the color difference at each pixel, e. g., $G - R$, $G - B$ or $B - R$ and to record the prediction error.

—**Statistical**. In this technique we exploit the distribution of selected data entities in the image. An example of such a technique is entropy coding. In this method a shorter codeword is assigned to data entries which are more likely to occur and to assign a longer codeword to data entries which are less likely to occur. Two realizations of entropy coding are Huffman coding and arithmetic coding.

For a high compression rate, we usually require more than one technique. Thus Chung and Chen's state-of-the-art algorithm [18] employs all three techniques: The input image is separated into four parts: G_0, G_1, R and B. The G_0 data is raster scanned and spatial prediction errors are calculated for each pixel. The G_0 prediction errors are then losslessly compressed using Huffman coding. Next a prediction of each G_1 pixel value is made using the G_0 data. If $\widehat{G}_1$ denotes the predicted G_1 value, then we losslessly compress the prediction errors $\delta G_1 = G_1 - \widehat{G}_1$ using Huffman coding. Finally we use the G_0 and G_1 data to predict the G value at each R and B pixel. If $\widehat{G}$ denotes the predicted G value, then we losslessly compress the color differences $\delta R = R - \widehat{G}$ and $\delta B = B - \widehat{G}$ using Huffman coding through spectral correlation.

Although this algorithm has state-of-the-art compression performance its implementation is very complicated because each color plane G_0, G_1, R and B is processed differently. Furthermore, its computational load is very high. A major reason for this is because each color plane uses a separate Huffman code. Finally the algorithm of Chung and Chen requires a requires a large memory buffer. For all of these reasons, lossless Bayer image compression applications in which the computational complexity and memory is limited, often use a switched prediction algorithm which we now describe in the next section.

3. SWITCHED PREDICTION ALGORITHMS

Switched predictive algorithms are widely used for lossless image compression [3, 4, 5, 20]. They are simple to implement, fast to run and have minimal memory requirements. In these algorithms a *gray-scale* image I is compressed as follows: The rows in I are scanned from top-to-bottom. Then in each row, the pixels are scanned from left-to-right. At each pixel (m, n) we classify the local pixel pattern, or context, $C(m, n)$ into one of K classes using the pixel values which have already been scanned. Then given the local class index $C(m, n)$ we predict the pixel value $I(m, n)$:

$$\hat{I}(m,n) = f(a,b,\ldots,g|k) \text{ if } C(m,n) = k \ , \qquad (1)$$

where $a, b, \ldots, g$, denote the nearest neigboring pixel values which have already been scanned; $f(a, b, \ldots, g|k), k \in \{1, 2, \ldots, K\}$, denotes an appropriate and fixed set of K prediction functions and $\hat{I}(m, n)$ denotes the predicted gray-value of the pixel (m, n). We then replace $I(m, n)$ by the prediction residual $\delta I(m, n) = \hat{I}(m, n) - I(m, n)$. If the predictions are accurate, the result is a residual image which is efficiently encoded using a variable-length Huffman, or arithmetic, entropy coder [21, 22].

In this article we consider three predictive lossless image compression algorithms.

—**JPEG-LS Algorithm** The JPEG-LS algorithm is a state-of-the-art standard developed by the Joint Photographic Experts Group (JPEG) for lossless compression of continuous tone still images [1]. In the JPEG-LS algorithm, we use the following $K = 3$ prediction functions:

$$\begin{aligned} f(a,b,\ldots,g|1) &= \min(a,b) \ , \\ f(a,b,\ldots,g|2) &= \max(a,b) \ , \\ f(a,b,\ldots,g|3) &= (a+b-c) \ , \end{aligned} \qquad (2)$$

where $a = I(m, n-1), b = I(m-1, n)$ and $c = I(m-1, n-1)$ are three neighboring previously scanned pixel values. The corresponding local context $C(m, n)$ is calculated using Eq. (3):

$$C(m,n) = \begin{cases} 1 \text{ if } c \geq \max(a,b) \ , \\ 2 \text{ if } c \leq \min(a,b) \ , \\ 3 \text{ otherwise } . \end{cases} \qquad (3)$$

—**Jiang Algorithm** The Jiang algorithm [6, 7] is a slightly more complicated generalization of the JPEG-LS algorithm and uses $K = 5$ prediction functions:

$$\begin{aligned} f(a,b,\ldots,g|1) &= \min(a,b) \ , \\ f(a,b,\ldots,g|2) &= (d+\min(a,b))/2 \ , \\ f(a,b,\ldots,g|3) &= \max(a,b) \ , \\ f(a,b,\ldots,g|4) &= (d+\max(a,b))/2 \ , \\ f(a,b,\ldots,g|5) &= (a+b-c) \ , \end{aligned} \qquad (4)$$

where $d = I(m-1, n+1)$. The local context may be calculated using the pseudo-code listed in Algorithm 1.

—**CALIC** The CALIC algorithm [4, 5] is substantially more complicated than both the JPEG-LS and the Jiang algorithms. It uses $K = 7$ prediction functions. Pseudo-code for calculating the local context $C(m, n)$ is given in Algorithm 2. The corresponding prediction functions are:

$$\begin{aligned} f(a,b,\ldots,g|1) &= a \ , \\ f(a,b,\ldots,g|2) &= (6a+2b+d-c)/8 \ , \\ f(a,b,\ldots,g|3) &= (10a+6b+3d-3c)/16 \ , \\ f(a,b,\ldots,g|4) &= b \ , \\ f(a,b,\ldots,g|5) &= (2a+6b+d-c)/8 \ , \\ f(a,b,\ldots,g|6) &= (6a+10b+3d-3c)/16 \ , \\ f(a,b,\ldots,g|7) &= (2a+2b+d-c)/4 \ , \end{aligned} \qquad (5)$$

where $e = I(m, n-2)$, $f = I(m-2, n)$ and $g = I(m-2, n+1)$.

Algorithm 1: Calculation of Jiang Context $C(m,n)$

```
Input: Previously transmitted pixel values $a = I(m-1.n)$,
       $b = I(m,n-1)$, $c = I(m-1,n-1)$,
       $d = I(m-1,n+1)$
Output: Context $C(m,n)$
if $c \geq \max(a,b)$ then
    if $(c - \max(a,b)) \geq 10$ & $d < b$ & $(a-b) \geq 5$ then
        $C(m,n) = 2$
    else
        $C(m,n) = 1$
else if $c \leq \min(a,b)$ then
    if
     $10 \leq (d-b) \leq 50$ & $|b-a| \leq 10$ & $(\min(a,b)-c) \geq 5)$
     then
        $C(m,n) = 4$
    else
        $C(m,n) = 3$
else
    $C(m,n) = 5$
```

Algorithm 2: Calculation of CALIC Context $C(m,n)$

```
Input: Previously transmitted pixel values $a = I(m,n-1)$,
       $b = I(m-1,n)$, $c = I(m-1,n-1)$,
       $d = I(m-1,n+1)$, $e = I(m,n-2)$,
       $f = I(m-2,n)$ and $g = I(m-2,n+1)$.
Output: Context $C(m,n)$
$dv = |a-c| + |b-f| + |d-g|$
$dh = |a-e| + |b-c| + |b-d|$
if $(dv - dh) > 80$ then
    $C(m,n) = 1$
else if $(dv - dh) < (-80)$ then
    $C(m,n) = 4$
else
    $C(m,n) = 7$
    if $(dv - dh) > 32$ then
        $C(m,n) = 2$
    else if $(dv - dh) > 8$ then
        $C(m,n) = 3$
    else if $(dv - dh) < (-32)$ then
        $C(m,n) = 5$
    else if $(dv - dh) < (-8)$ then
        $C(m,n) = 6$
```

3.1 De-interlacing

The basic idea underlying the lossless prediction algorithms is that in a continuous-tone gray-scale image, the pixel values vary smoothly as we move from one pixel to the neighboring pixel. In this case, the predictions are generally accurate and thus the prediction residuals may be efficiently encoded. Unfortunately, this is not true for the Bayer image: neighboring pixels represent different colors and so as we move from one pixel to the next we observe a substantial change in the pixel value. This in turn adversely affects the compression rate.

Let us denote the input Bayer image as $\boldsymbol{I}$ which we regard as consisting of four component images (each of size $M \times N$) which we denote as G_0, R, B and G_1 as shown in Fig. 1. If (i,j) denotes a pixel in the Bayer image $\boldsymbol{I}$ and (m,n) denotes the corresponding pixel in one of the component images, then

G_0	R	G_0	R
B	G_1	B	G_1
G_0	R	G_0	R
B	G_1	B	G_1

Fig. 1. Bayer color filter array. This consists of two green components (G_0 and G_1), one red component (R) and one blue component (B).

$$\boldsymbol{I}(i,j) = \begin{cases} G_0(m,n) & \text{if } m = \lfloor (i+1)/2 \rfloor, n = \lfloor (i+1)/2 \rfloor , \\ R(m,n) & \text{if } m = \lfloor (i+1)/2 \rfloor, n = \lfloor j/2 \rfloor , \\ B(m,n) & \text{if } m = \lfloor i/2 \rfloor, n = \lfloor (j+1)/2 \rfloor , \\ G_1(m,n) & \text{if } m = \lfloor i/2 \rfloor, n = \lfloor j/2 \rfloor . \end{cases} \tag{6}$$

where $\lfloor x \rfloor$ denotes the value of x rounded down to the nearest integer.

To efficiently compress the Bayer image $\boldsymbol{I}$ we may de-interlace $\boldsymbol{I}$ by *physically* separating $\boldsymbol{I}$ into its component parts G_0, R, B and G_1 and then apply the JPEG-LS, Jiang or CALIC compression algorithms to each component separately. An equivalent, but much simpler, approach [3] is to define a new set of *interleaved* nearest neighbor pixels: $\{a_{new}, b_{new}, \ldots, g_{new}\}$, where

$$\begin{aligned} a_{new} &= \boldsymbol{I}(i, j-2) , \\ b_{new} &= \boldsymbol{I}(i-2, j) , \\ c_{new} &= \boldsymbol{I}(i-2, j-2) , \\ d_{new} &= \boldsymbol{I}(i-2, j+2) , \\ e_{new} &= \boldsymbol{I}(i, j-4) , \\ f_{new} &= \boldsymbol{I}(i-4, j) , \\ g_{new} &= \boldsymbol{I}(i-4, j+2) . \end{aligned} \tag{7}$$

In this case, the pixel $\boldsymbol{I}(i,j)$ and the interleaved nearest neighbors $\{a_{new}, b_{new}, \ldots, g_{new}\}$ all belong to the *same* component image (see Fig. 2). We then define *interleaved* JPEG-LS, Jiang and CALIC algorithms by replacing $\{a, b, \ldots, g\}$ in Eqs. (2-5) and in Algorithms 1 and 2 by $\{a_{new}, b_{new}, \ldots, g_{new}\}$. To compress $\boldsymbol{I}$ we then simply apply the interleaved JPEG-LS, Jiang or CALIC algorithms *directly* to $\boldsymbol{I}$ [3].

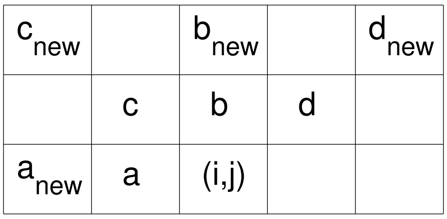
It is important to note, however, that notwithstanding the separation of $\boldsymbol{I}$ into its component parts, we still only use *one* Huffman code to encode the residuals $\delta\boldsymbol{I}(m,n)$. This is, of course, possible because the components G_0, R, B and G_1 all have the same dynamic range and the same statistical characteristics.

3.2 Decorrelation of the Color Planes

In order to further improve the performance of the JPEG-LS, Jiang and CALIC algorithms we may, after de-interleaving, decorrelate the G_0, G_1, R and B planes. Let U, V, W and X denote the

Table 1. Lossless Image Compression Rates (bits/pixel)

Img No	JPEG-LS			Jiang			CALIC		
	Original	De-interleave	IC^3	Original	De-interleave	IC^3	Original	De-interleave	IC^3
01	6.812	6.245	6.103	6.816	6.251	6.030	6.724	6.155	$\underline{6.027}$
02	7.004	5.006	4.871	7.005	5.012	4.798	7.178	4.903	$\underline{4.755}$
03	6.867	4.661	4.433	6.866	4.665	$\underline{4.315}$	6.810	4.579	4.349
04	7.547	5.263	5.086	7.547	5.266	$\underline{4.941}$	7.483	5.138	4.996
05	6.847	6.604	6.255	6.851	6.615	$\underline{6.115}$	6.788	6.515	6.167
06	6.812	5.847	5.708	6.816	5.854	5.642	6.724	5.764	$\underline{5.608}$
07	7.004	4.846	4.659	7.000	4.845	$\underline{4.528}$	7.178	4.874	4.549
08	6.866	6.235	6.135	6.866	6.405	6.096	6.810	6.189	$\underline{6.027}$
09	7.547	4.880	4.747	7.547	4.884	$\underline{4.630}$	7.483	4.792	4.646
10	6.847	4.935	4.802	6.851	4.937	$\underline{4.684}$	6.788	4.882	4.694
11	6.592	5.535	5.394	6.592	5.543	5.323	6.485	5.470	$\underline{5.298}$
12	6.593	4.835	4.688	6.593	4.837	4.606	6.510	4.764	$\underline{4.601}$
13	7.098	6.967	6.793	7.102	6.973	$\underline{6.672}$	6.990	6.846	6.723
14	6.951	6.072	5.869	6.951	6.082	$\underline{5.747}$	6.893	5.977	5.767
15	7.112	4.988	4.873	7.114	4.989	$\underline{4.720}$	6.997	4.851	4.759
16	6.064	5.225	5.089	6.065	5.231	5.013	5.868	5.171	$\underline{4.990}$
17	5.469	5.225	5.034	5.468	5.227	$\underline{4.872}$	5.401	5.121	4.931
18	6.713	6.138	5.967	6.713	6.144	$\underline{5.833}$	6.651	6.017	5.880
19	6.223	5.489	5.347	6.218	5.496	$\underline{5.244}$	6.213	5.362	5.255
20	5.926	4.371	4.222	5.926	4.374	$\underline{4.145}$	5.899	4.301	4.169
21	6.305	5.637	5.518	6.306	5.642	$\underline{5.410}$	6.266	5.551	5.433
22	6.938	5.587	5.467	6.938	5.593	$\underline{5.357}$	6.824	5.480	5.381
23	7.779	4.565	4.449	7.779	4.560	$\underline{4.290}$	7.705	4.403	4.324
24	6.247	5.850	5.631	6.247	5.853	$\underline{5.522}$	6.140	5.819	5.573
ave	6.637	5.459	5.298	6.638	5.463	$\underline{5.189}$	6.552	5.372	5.204



c_{new}		b_{new}		d_{new}
	c	b	d	
a_{new}	a	(i,j)		

Fig. 2. The Bayer image $\boldsymbol{I}$ and a pixel (i, j) and the previously scanned neighbors $\{a, b, c, \ldots, g\}$ and $\{a_{new}, b_{new}, c_{new}, \ldots, g_{new}\}$, where $a = \boldsymbol{I}(i, j-1)$, $b = \boldsymbol{I}(i-1, j)$, $c = \boldsymbol{I}(i-1, j-1)$, $d = \boldsymbol{I}(i-1, j+1)$, $e = \boldsymbol{I}(i, j-2)$, $f = \boldsymbol{I}(i-2, j)$, $g = \boldsymbol{I}(i-2, j+1)$ and $a_{new} = \boldsymbol{I}(i, j-2) = e$, $b_{new} = \boldsymbol{I}(i-2, j) = f$, $c_{new} = \boldsymbol{I}(i-2, j-2)$, $d_{new} = \boldsymbol{I}(i-2, j+2)$, $e_{new} = \boldsymbol{I}(i, j-4)$, $f_{new} = \boldsymbol{I}(i-4, j)$, $g_{new} = \boldsymbol{I}(i-4, j+2)$.

corresponding de-correlated planes. Mohammed and Wahid [23] have shown how this process may be simply and efficiently implemented. Unfortunately, although color plane decorrelation does increase the compression ratio there is also a substantial increase in algorithm complexity and processing time. The reason for this is that after decorrelation, the decorrelated planes U, V, W, X have different dynamic ranges and different statistical characteristics. This means we must use four different Huffman codes: one for each of the decorrelated planes. This, in turn, leads to a substantial increase in both algorithm complexity and processing time.

4. INTER-COLOR CONTEXT CORRELATION IC^3 CLASSIFIER

In this we article we describe a new approach to improving the performance of the JPEG-LS, Jiang and CALIC algorithms. The idea is to replace the individual context classifiers by a new high performance and *generic* Inter-Color Context (IC^3) Classifier.

The idea behind the IC^3 classifier is that pixels which are physically close to one another, should have similar local contexts *irrespective* of the pixel color. Employing this idea we replace the JPEG-LS, Jiang and CALIC context classifiers (Eq. 3 and Algorithms 1 and 2) with the new *generic* context classifier (IC^3), $C_{new}(i, j)$, and thus obtain a higher lossless Bayer image compression.

Mathematically we formulate the IC^3 classifier as follows. Let (i, j) denote a pixel in $\boldsymbol{I}$ and suppose $(i^{(p)}, j^{(p)}), p \in \{1, 2, \ldots, P\}$, denote P nearby and previously scanned pixels. Since the $(i^{(p)}, j^{(p)})$ are previously scanned pixels, we may calculate K different pixel residues $\delta\boldsymbol{I}(i^{(p)}, j^{(p)}|k), k \in \{1, 2, \ldots, K\}$, for each $(i^{(p)}, j^{(p)})$:

$$\delta\boldsymbol{I}(i^{(p)}, j^{(p)}|k) = f(a^{(p)}_{new}, b^{(p)}_{new}, \ldots |k) - \boldsymbol{I}(i^{(p)}, j^{(p)}), \quad k \in \{1, 2, \ldots, K\} \ . \tag{8}$$

Fig. 3. Set of 24 Bayer Kodak test images. The images are generated by applying the Bayer CFA to the corresponding Kodak test images. Note the characteristic large changes in intensity in the Bayer images as we move from one pixel to its neighbor.

We now aggregate the absolute residuals $|\delta\boldsymbol{I}(i^{(p)},j^{(p)}|k)|, p \in \{1,2,\ldots,P\}$, using an aggregation operator $\widehat{\theta}$:

$$\Delta\boldsymbol{I}(k) = \widehat{\theta}\big(|\delta\boldsymbol{I}(i^{(1)},j^{(1)}|k)|, |\delta\boldsymbol{I}(i^{(2)},j^{(2)}|k)|, \ldots, |\delta\boldsymbol{I}(i^{(p)},j^{(p)}|k)|\big); \qquad (9)$$

Then we define an *optimum* local context for the pixel (i,j) as $C^*_{new}(i,j) \equiv k^*$, where

$$k^* = \arg\min_{k=1}^{K} \Delta\boldsymbol{I}(k) \; . \qquad (10)$$

There are many possible aggregation operators $\widehat{\theta}$. Experimentally, we found the best results were obtained using a $\widehat{\theta} = \max$ operator. The corresponding optimum local context for the pixel (i,j) is thus $C^*_{new}(i,j) = k^*$, where

$$k^* = \arg\min_{k=1}^{K}\left(\max_{p=1}^{P}\big(|\delta\boldsymbol{I}(i^{(p)},j^{(p)}|k)|\big)\right) \; . \qquad (11)$$

Algorithm 3 summarizes the processing steps in the new IC^3 classifier.

We now summarize the main steps in losslessly compressing a Bayer image using the IC^3 context classifier and a set of interleaved prediction functions.

(1) Scan the pixels in the Bayer image top-to-bottom, left-to-right.

(2) Suppose (i,j) is the current pixel and $(i^{(p)},j^{(p)}), p \in \{1,2,\ldots,P\}$, denote the $P = 3$ nearest previously scanned pixels. For each pixel $(i^{(p)},j^{(p)}), p \in \{1,2,\ldots,P\}$, calculate the pixel residual errors $\delta\boldsymbol{I}(i^{(p)},j^{(p)}|k)$, $k \in \{1,2,\ldots,K\}$, using Eq. (8).

```
Algorithm 3: Calculation of $IC^3$ Context $C_{new}(i,j)$
Input: Bayer image $\boldsymbol{I}$.
Output: $IC^3$ Context Classifier $C_{new}(i,j)$
Let $(i,j)$ be a pixel in $\boldsymbol{I}$ and let $(i^{(p)},j^{(p)}), p\{1,2,\ldots,P\}$, be
the $P$ previously scanned pixels nearest to $(i,j)$.
Use Eq. (8) to calculate $K$ pixel residuals for each pixel
$(i^{(p)},j^{(p)}), p \in \{1,2,\ldots,P\}$:
$\delta\mathbf{I}(i^{(p)},j^{(p)}|k) = f(a^{(p)}_{new}, b^{(p)}_{new}, \ldots, |k) - \boldsymbol{I}(i^{(p)},j^{(p)})$,
  $k \in \{1,2,\ldots,K\}$.
Use Eq. (11) to calculate the optimum context
$C_{new}(i,j) \equiv k^*$ for the pixel $(i,j)$.
```

(3) Aggregate the $|\delta\boldsymbol{I}(i^{(p)},j^{(p)}|k)|$ using the $\hat{\theta} = max$ operator according to Eq. (9).

(4) Use Eq. (11) to calculate the optimum context k^* for (i,j).

(5) Calculate the predicted gray-level for the pixel (i,j) using the context k^* and the interleaved prediction function $f(a_{new}, b_{new}, \ldots |k^*)$.

(6) Losslessly encode the prediction error $\delta\boldsymbol{I}(i,j|k^*)$ using a variable-length Huffman, or arithmetic entropy coder, where

$$\delta\boldsymbol{I}(i,j|k^*) = f(a_{new}, b_{new}, \ldots |k^*) - \boldsymbol{I}(i,j) \; , \qquad (12)$$

5. RESULTS

We measured the performance of the new IC^3 classifier on the standard set of 24 Kodak RGB images. First the set of Kodak im-

Table 2. Lossless Image Compression Rates as Percentage of Original JPEG-LS Compression

Img No		JPEG-LS			Jiang			CALIC	
	Original	De-interleave	IC^3	Original	De-interleave	IC^3	Original	De-interleave	IC^3
01	100.0	91.7	89.6	100.1	91.8	88.5	98.7	90.4	88.5
02	100.0	71.5	69.5	100.0	71.6	68.5	102.5	70.0	67.9
03	100.0	67.9	64.6	100.0	67.9	62.8	99.2	66.7	63.3
04	100.0	69.7	67.4	100.0	69.8	65.5	99.2	68.1	66.2
05	100.0	96.5	91.4	100.1	96.6	89.3	99.1	95.1	90.1
06	100.0	84.3	82.2	100.0	84.4	81.3	98.0	83.0	80.8
07	100.0	72.2	69.4	100.0	72.2	67.5	96.8	72.6	67.8
08	100.0	95.6	94.1	100.1	95.8	93.5	98.8	94.9	92.5
09	100.0	81.5	79.3	100.0	81.6	77.3	97.8	80.0	77.6
10	100.0	81.5	79.3	100.0	81.5	77.4	96.3	80.6	77.5
11	100.0	84.0	81.8	100.0	84.1	80.8	98.4	83.0	80.4
12	100.0	73.3	71.1	100.0	73.4	69.9	98.7	72.3	69.8
13	100.0	98.2	95.7	100.0	98.2	94.0	98.5	96.4	94.7
14	100.0	87.4	84.4	100.0	87.5	82.7	99.2	86.0	83.0
15	100.0	70.1	68.5	100.0	70.2	66.4	98.4	68.2	66.9
16	100.0	86.2	83.9	100.0	86.3	82.7	96.8	85.3	82.3
17	100.0	95.5	92.0	100.0	95.6	89.1	98.8	93.6	90.2
18	100.0	91.4	88.9	100.0	91.5	86.9	99.1	89.6	87.6
19	100.0	88.2	85.9	99.9	88.3	84.3	99.8	86.2	84.4
20	100.0	73.8	71.3	100.0	73.8	69.9	99.5	72.6	70.4
21	100.0	89.4	87.5	100.0	89.	85.8	99.4	88.0	86.2
22	100.0	80.5	78.8	100.0	80.6	77.2	98.4	79.0	77.6
23	100.0	58.7	57.2	100.0	58.6	55.1	99.1	56.6	55.6
24	100.0	93.6	90.1	100.0	93.7	88.4	98.3	93.1	89.2
ave	100.0	82.6	80.2	100.1	82.7	78.5	98.7	81.3	78.8

ages are converted to Bayer images by applying the Bayer CFA (Fig. 1) to each Kodak image. The corresponding Bayer Kodak images are shown in Fig. 3.

The Bayer Kodak images were then losslessly compressed using the JPEG-LS, Jiang and CALIC algorithms under three different conditions.

(1) **Original**. The images were losslessly compressed *without* de-interleaving using the JPEG-LS, Jiang and CALIC algorithms (i.e. standard non-interleaved JPEG-LS, Jiang and CALIC prediction functions and the corresponding non-interleaved context classifiers).

(2) **De-Interleave**. The images were losslessly compressed *after* de-interleaving and using the JPEG-LS, Jiang and CALIC algorithms (i.e. interleaved JPEG-LS, Jiang and CALIC prediction functions and the corresponding interleaved context classifiers).

(3) IC^3. The images were losslessly compressed *after* de-interleaving and using the new IC^3 context classifier (i.e. interleaved JPEG-LS, Jiang and CALIC predictions functions and the new generic IC^3 context classifier).

The bit-rates are given in Table 1 and the corresponding percentage improvements in Table 2. For each image, the algorithm with the lowest bit-rate is underlined. Note that in each experiment only *one* variable-length Huffman code was used.

We observe that for each Bayer Kodak test image and each set of prediction functions the new IC^3 classifier gave the smallest bit-rate. For most of the test images (17 out of 24) the lowest bit-rates was obtained using the interleaved Jiang prediction functions and the new IC^3 classifier. In the remaining test images, the lowest bit-rate was obtained using the interleaved CALIC prediction functions and the new IC^3 classifier. Moreover, in these test images, the difference between the Jiang/IC^3 and the CALIC/IC^3 algorithms was minimal. Next we compared the average bit-rates obtained with the different algorithms. Comparing the Jiang/IC^3 algorithm with the standard non-interleaved JPEG-LS algorithm and the de-interleaved JPEG-LS algorithms we obtain reductions of ~ 1.5 bit/pixel ($\sim 21.5\%$) and ~ 0.27 bit/pixel ($\sim 4.1\%$).

6. CONCLUSION

We have described a new *generic* Inter-Color Context Classifier (IC^3) which may be used in any switched prediction algorithm for lossless Bayer image compression. For each set of prediction functions (JPEG-LS, Jiang and CALIC), employing the IC^3 classifier gave a substantial reduction in bit-rate for each of the 24 Bayer Kodak test images. This is brought out in Fig. 4.

For each Bayer Kodak test image we ranked the bit-rates in Table 1, where the lowest bit-rate was assigned a rank of 1 and the highest bit-rate was assigned a rank of 9. Fig. 4 shows the average rank obtained for each compression algorithm. We see clearly how the IC^3 classifier lowers the average bit-rate. In particular, Jiang/IC^3 combination had the lowest average rank followed by CALIC/IC^3 and then JPEG-LS/IC^3.

The new context classifier only requires a simple de-interleaving pre-processing stage. Most importantly, the new context classifier does *not* require the use of multiple Huffman, or arithmetic entropy, codes. The new classifier is thus simple to implement, fast to run and has a low memory use.

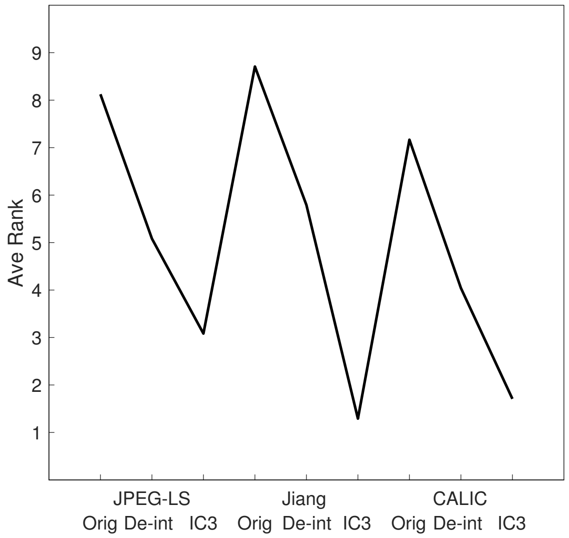


Fig. 4. Average rank for each compression algorithm. For each Bayer Kodak test image we ranked the bit-rates listed in Table 1, where the lowest bit-rate was assigned a rank of 1 and the highest bit-rate was assigned a rank of 9.

7. REFERENCES

[1] M. J. Weinberger, G. Seroussi, G. Sapiro, Loco-I lossless image compression algorithm: Principles and standardization into jpeg-ls, IEEE Trans Image Process 9 (2000) 1309–1326.

[2] A. Bazhyna, A. Gotchev, K. Egiazarian, Lossless compression of Bayer pattern color filter arrays, in: Proceedings SPIE, volume 5672, 2005, pp. 378–387.

[3] A. Trifan, A. J. R. Neves, A survey on lossless compression of Bayer color filter array images, Int J Computer Inform Engineer 10 (2016) 729–734.

[4] N. Memon, D., X. Wu, Recent developments in context-based predictive techniques for lossless image compression, Compter J. 40 (1997) 127–136.

[5] X. Wu, N. D. Memon, Context-based, adaptive, lossless image coding, IEEE Trans Commun 45 (1997) 437–444.

[6] C.-C. Chang, G.-I. Chen, Enhancement algorithm for non-linear context-based predictors, IEE Proc Vision Image and Signal Process 150 (2003) 15–19.

[7] J. Jiang, B. Guo, S. Y. Yang, Revisiting the JPEG-LS prediction scheme, IEE Proc Vis Image Signal Process 20 (2000) 575–580.

[8] X. Li, Demosaicking by successive approximation, IEEE Trans Image Process 14 (2005) 370–379.

[9] D. Kiku, M. Monno, Y.and Tanaka, M. Okutomi, Beyond color difference: Residual interpolation for color image demosacking, IEEE Trans Image Process 25 (2016) 1288–1300.

[10] D. Menon, G. Calvagno, Color image demosaicking: an overview, Image Communication 26 (2011) 518–533.

[11] M. Hernandez-Cabronero, I. Blanes, F. Auli-Llinas, M. W. Marcellin, , J. Serra-Sagrista, Mosaic-based color transform optimization for lossy and lossy-to-lossless compression of pathology whole-slide images, IEEE Trans Medical Imaging 38 (2019) 21–32.

[12] M. Hernandez-Cabronero, M. W. Marcellin, I. Blancs, I. Serra-Sagrista, Lossless compression of color filter array mosaic images with visualization via jpeg 2000, IEEE Trans Multimedia 20 (2018) 257–270.

[13] S. K. Mohammed, K. M. Mafijur Rahman, K. A. Wahid, Lossless compression in Bayer color filter array for capsule endoscopy, IEEE Access 5 (2017) 13823–13834.

[14] A. Bazhyna, Image compression in digital cameras, in: PhD thesis, Tampere University of Technology, Finland, (2009)

[15] M.W. Alam, M.M. Hasan, S.K. Mohammed, F. Deeba, K.A. Wahid, Are current advances of compression algorithms for capsule endoscopy enough? a technical review, IEEE Reviews in Biomedical Engineering 10 (2017) 26–43.

[16] N. Lian, L. Chang, V. Zagorodnov, Y. Tan, Reversing demosaicking and compression in colour filter array image processing: performance analysis and modeling, IEEE Trans Image Process 15 (2006) 3261–3278.

[17] S.-Y. Lee, A. Ortega, A novel approach for image compression in digital cameras with a bayer color filter array, in: IEEE Int Conf Image Process (ICIP'01), volume 3, 2001, pp. 482–485.

[18] K.-H. Chung, Y.-H. Chen, A lossless compression scheme for Bayer color filter array images, IEEE Trans Image Process 17 (2008) 134–144.

[19] N. Zhang, X. Wu, Lossless compression of color mosaic images, IEEE Trans Image Process 15 (2005) 1379–1388.

[20] S. Marusic, G. Deng, Adaptive prediction for lossless image compression, Image Communication 17 (2002) 363–372.

[21] S. W. Golomb, Run-length encodings, IEEE Trans Inf Theory 12 (1966) 399–401.

[22] G. G. Langdon Jr, An introduction to arithmetic coding, IBM J Res Develop 28 (1984) 135–149.

[23] S. K. Mohammed, K. A. Wahid, Lossless and reversible color space transformation for Bayer color filter array images, IEE Image Process 12 (2018) 1485–1490.