Lossless Compression of HDR Color Filter Array Image for the Digital Camera Pipeline

Dohyoung Lee, Konstantinos N. Plataniotis

Multimedia Lab, The Edward S. Rogers Department of Electrical and Computer Engineering, University of Toronto, 10 Kings College Road, Toronto, Canada

Abstract

This paper introduces a lossless color filter array (CFA) image compression scheme capable of handling high dynamic range (HDR) representation. The proposed pipeline consists of a series of pre-processing operations followed by a JPEG XR encoding module. A deinterleaving step separates the CFA image to sub-images of a single color channel, and each sub-image is processed by a proposed weighted template matching prediction. The utilized JPEG XR codec allows the compression of HDR data at low computational cost. Extensive experimentation is performed using sample test HDR images to validate performance and the proposed pipeline outperforms existing lossless CFA compression solutions in terms of compression efficiency.

Keywords: color filter array, digital camera pipeline, lossless compression, high dynamic range, weighted template matching prediction

1. Introduction

In digital cameras, color information of a real-world scene is acquired through an image sensor, usually a charge-coupled device (CCD) [1] or a complementary metal oxide semiconductor (CMOS) [2] sensor in the format of superimposition of three primary colors, red (R), green (G), and blue (B). Commonly used image sensors are monochromatic devices that sense the light within limited frequency range, and thus cannot record color information directly. To reduce production cost and complexity, most consumer level digital cameras exploit a single-sensor imaging technology, which captures a visual scene in color using a monochrome sensor in conjunction with a color filter array (CFA) [3]. A CFA is a mosaic of color filters placed on the top of conventional CCD/CMOS image sensors to filter out two of the RGB components in each pixel position. Since the image acquired through CFA only contains a single color component at each cell, the CFA image initially appears as an interleaved mosaic similarly to a grayscale image. Color demosaicking (CDM) estimates missing two components in each pixel location of the CFA image to produce the full-color image.

In a conventional digital camera pipeline, CDM is initially performed on the CFA image, followed by compression of demosaicked image. Due to its simple user interface and device compatibility, the CDM first pipeline became a dominant workflow in digital camera designs. However CDM increases the size of the data by a factor of 3, ultimately leading to inefficient usage of storage memory, despite the application of compression. As an alternative solution to the conventional approach, a compression-first pipeline has been introduced. This approach compresses the CFA image prior to applying a CDM, requiring less computational resource and storage capacity as a number of CFA samples are only 1/3 of ones in the full RGB images. In addition, it allows the user to acquire high quality images by performing sophisticated CDMs on the end devices, such as personal computers, where sufficient processing power is provided. The most straightforward implementation of the compression-first approach is a direct application of standard image compression tools, such as JPEG [4] and JPEG 2000 [5], on raw CFA images. However, it is found to be inefficient since intermixing pixels from different color channels generates artificial discontinuity whereas compression solutions are generally optimized for continuous tone images. In order to address this issue, advanced CFA compression schemes typically exploit various pre-processing operations for optimal use of compression tools.

The prior art CFA compression schemes are gener-
ally categorized in two types: lossy [6–12] and lossless [13–16], depending on nature of pre-processing algorithms and compression tools. Lossy approaches aim to minimize amount of image data by discarding visually redundant contents. They are suitable for the areas where the efficient usage of memory and computational resource is paramount. On the other hands, lossless compression is crucial in the field of medical imaging, cinema industry, and image archiving system of museum arts and relics, where the exact replica of the original data is preferred over high compression ratio. Lossless CFA compression schemes typically deinterleave color channels [15, 16] or perform wavelet decomposition [13] prior to applying an encoding operation to alleviate the aliasing issue in the direct CFA encoding method.

In this paper, we present a new lossless CFA compression method that encodes a Bayer CFA image [17]. We focus on the Bayer CFA structure as it is the dominant CFA arrangement in the industry. The proposed scheme consists of color channel deinterleave, weighted template matching prediction, and lossless image compression operations.

There are two main differences of the proposed method compared to prior art CFA image compression solutions. First of all, we make use of the JPEG XR compression standard [18] to facilitate compression of CFA image in high bit-depth/high dynamic range (HDR) representation. HDR images require a higher number of bits per color channel than traditional 8 bit images to allow realistic representation of the scene with smoother tonal gradation from shadow to highlight. Due to the increasing popularity of HDR photography, digital camera manufacturers started to offer a HDR capture mode in their cameras that produces HDR CFA data in a raw format, typically between 10 and 16 bit per pixel (bpp), via high-end sensors or combined exposure operations. Although storage of raw images allows the user to retain the necessary high bit-depth data for the full range of post-processing operations, it leads to excessive usage of memory resources due to absence of compression operation. To the best of author’s knowledge, most previous works in CFA compression are limited to codecs applied to conventional 8 bit input (except for a YDgCoG transform based scheme [19] in HD Photo codec [20] capable of supporting greater than 8 bit CFA data). Such a conventional pipeline causes a loss of precision in HDR CFA data as the original HDR data stream is mapped onto a 8 bit equivalent representation prior to applying compression solution. Among various codecs capable of handling HDR input, we believe that JPEG XR’s balance between performance and complexity makes it a suitable solution for digital camera implementation. Secondly, we introduce a predictive coding based lossless CFA compression scheme that employs a weighted template matching predictor to increase the accuracy of pixel prediction and achieve high compression efficiency. Our predictor is similar to the context matching based prediction (CMBP) presented in [16] in the sense that CFA image is separated into green and non-green sub-images, and each sub-image undergoes predictive coding procedure by computing weighted sum of neighbor pixels based on template matching. However, our proposed method allows for more precise prediction of pixel values due to its adaptive weight computation scheme whereas the CMBP uses pre-defined weight coefficients based on ranking of neighbor pixels.

The rest of paper is structured as follows. Section 2 studies background information behind the proposed scheme. Section 3 presents the proposed lossless CFA compression scheme in detail. Experiment results and analysis are demonstrated in Section 4 and conclusion is given in Section 5.

2. Background

This section provides background information necessary for an understanding of the proposed algorithm.

2.1. JPEG XR compression standard

JPEG XR (extended range) is a recently introduced image compression codec and the latest member of JPEG standards family derived from Microsoft HD Photo [21]. JPEG XR offers wide range of input bit-depth support from 1 bit through 32 bit per component. 8-bit and 16-bit formats are supported for both lossy and lossless compression, while 32-bit format is only supported for lossy compression since only 24 bits are typically retained through internal operations. Following the traditional image compression structure, JPEG XR’s coding path includes color space conversion, block transform called Lapped Bi-orthogonal Transform (LBT), quantization, and entropy coding. The LBT is based on two basic operators: i) Picture Core Transformation (PCT) that decorrelates signal within a block region, and ii) Picture Overlapped Transformation (POT) which applies filtering across block edges to prevent blocking artifacts. As a results of the LBT the image data is converted from spatial domain to frequency domain, by producing three frequency sub-bands, DC (Direct Current), LP (Low Pass), and HP (High Pass). The LBT is a fully reversible operation, and thus allows lossless compression of image data.
JPEG XR provides many convenient features offered in its ancestor JPEG 2000 while maintaining its architecture considerably simpler than JPEG 2000 since it only uses integer based computations internally. Due to its wide dynamic range support, high rate-distortion performance, rich feature sets and efficiency codec architecture, JPEG XR is highly suitable for small, low powered consumer electronics.

2.2. Predictive coding

An algorithm called predictive coding, or differential coding, is widely used in lossless compression standards to compress image data without introducing any distortions. In predictive coding, the compression is generally carried out in two phases.

(i) Prediction of pixels: Pixels in the image are processed in a fixed order, usually in a raster scan order, row by row and left to right within a row. In this phase, the current pixel is predicted from the preceding pixels which have already been encoded. Since spatially adjacent pixels in a natural image are highly correlated to each other, a pixel can be predicted with a good accuracy from its neighboring pixels.

(ii) Encoding of prediction error signal: A predicted value is subtracted from the original value of the corresponding pixel to generate a prediction error, or residue. The prediction error signal is then entropy coded to minimize the data size.

The performance of predictive coding is substantially affected by the accuracy of prediction algorithm. If the predictor is well designed, the distribution of prediction error will be closely concentrated on zero and the variance of the prediction error will be much lower than that of the original signal, leading to improved compression efficiency.

A number of different predictors have been proposed for compression of images. A popular lossless image compression standard, JPEG-LS [22] exploits a predictor called Median Edge Detector (MED) which provides a good balance between prediction accuracy and computational simplicity. It predicts the value of the current pixel by examining 3 neighboring pixels in the directions of north, west, and north-west. Another image codec CALIC [23] employs an advanced predictor called Gradient Adaptive Predictor (GAP) that performs prediction based on local gradient information, estimated using 7 neighboring pixels.

There exist several linear predictors which predict the intensity of each pixel by weighted sum of its candidate pixels (neighboring pixels that have already been encoded). For a given pixel \( p_0 \), the prediction value \( \hat{p}_0 \) obtained by a linear predictor is given by:

\[
\hat{p}_0 = \sum_{k=1}^{K} w_k \cdot p_k
\]

where \( p_k \) is the candidate pixel, \( w_k \) is the weight coefficient associated with the candidate pixel \( p_k \), and \( K \) is the number of candidate pixels (also represents an order of prediction). Pixels are individually processed by encoding the prediction error \( e = p_0 - \hat{p}_0 \). The prediction error \( e \) is the only information sent to the decoder for the reconstruction of the input image as only previously encoded pixels are used in prediction. For lossless encoding, it is essential that both encoder and decoder produce identical prediction values.

Activity Level Classification Model (ALCM) [24] is a widely used adaptive linear predictors. In ALCM predictor, equal weight coefficients are initially given to all candidate pixels, and only largest and smallest weights are adjusted depending on the prediction result of the previously encoded pixel. Edge-Directed Prediction (EDP) [25] takes into account local structure information of the image (such as edges or textures) to calculate the weighting of candidate pixels. Some advanced schemes such as the Variable Blocksize Prediction [26] offer superior prediction accuracy, but requires side information (i.e. encoding parameters) to be transmitted to decoder side along with the prediction error. Therefore, there is a trade-off between the data overhead due to the side information and the degree of increased accuracy.

3. Proposed Algorithm

Fig. 1 illustrates the proposed CFA compression method for encoding process and decoding process. The proposed scheme employs a structure separation to extract 3 sub-images of single color component from the original CFA layout. Then, each sub-image undergoes a predictive coding process. The predictive coding forms a prediction for current pixel based on a linear combination of previously coded neighborhood pixels, and encodes the prediction error signal to remove spatial redundancies. Initially we process G sub-image using the weighted template matching prediction technique in raster scan order, and generate the prediction error of G channel, \( e_G \). After completion of G channel prediction, non-green sub-images are handled in similar manner. Instead of carrying out the prediction on R and B samples directly, we use color difference domain signals, \( dr \) (G-R), and \( db \) (G-B) for non-green compo-
nents. This allows us to reduce spectral (inter-channel) redundancies in the data, leading to higher compression efficiency. In order to obtain color difference signals, the estimation of missing G values at non-green pixel positions is necessary. In the proposed algorithm, we perform a bilinear interpolation on a quincunx G sub-image, which delivers satisfactory performance at low computational cost. Again, the prediction error of color difference signals, $e_d$ and $e_b$, are obtained by the proposed predictor. The generated error signals constitute standard 4:2:2 formatted data. Therefore, they are encoded by JPEG XR codec using its 4:2:2 lossless encoding mode.

In the companion decoding pipeline, compressed prediction error signals are decoded. Then the decoder forms the identical prediction as the one from the encoding pipeline using decompressed error signal to reconstruct individual sub-images. Finally, we combine generated sub-images to reconstruct original CFA layout.

3.1. Deinterleaving Bayer CFA

The proposed scheme initially deinterleaves the Bayer CFA images into three sub-images, as shown in Fig. 2. As previously mentioned, the direct compression of CFA image is inefficient as CFA data are formed by intermixing samples from different color channels. Although for most natural images, there still exist spatial correlations between CFA samples, pixels from different color channels create high frequency discontinuities,
disallowing high compression ratio. By deinterleaving the CFA image, three downsampled sub-images, each of which consists of pixels in a single color channel, are extracted.

Let us consider, a $K_1 \times K_2$ grayscale CFA image $z_{(i,j)} : Z^2 \rightarrow Z$ representing a two-dimensional input image to encode. The deinterleaving process can be formulated as follows:

$$
g_{(i,j)} = \begin{cases} z_{(i,j)}, & (i,j) \in \{(2m-1,2n),(2m,2n-1)\} \\
0, & \text{otherwise}
\end{cases}
$$

$$
r_{(i,j)} = \begin{cases} z_{(i,j)}, & (i,j) \in \{(2m-1,2n-1)\} \\
0, & \text{otherwise}
\end{cases}
$$

$$
b_{(i,j)} = \begin{cases} z_{(i,j)}, & (i,j) \in \{(2m,2n)\} \\
0, & \text{otherwise}
\end{cases}
$$

where $m = 1,2,\cdots,K_1/2$, and $n = 1,2,\cdots,K_2/2$. The obtained R and B sub-images form square lattices, while the obtained G sub-image constitutes a quincunx lattice. Each sub-image contains pixels from same color component and thus, subsequent prediction process can effectively remove spatial redundancies to achieve high compression performance.

3.2. Green sub-image prediction

The compression efficiency of predictive coding depends on the accuracy of a prediction model. Simple linear predictors often yield poor performance at image edge regions. The proposed adaptive predictor exploits a template matching technique to achieve high prediction performance. It measures the dissimilarity between the template of a current pixel to predict and the template of candidate pixels in neighbor to determine weight factors of candidate pixels. The weight factors adaptively increases the influence of candidate pixel whose associated template closely resembles the template of the pixel to predict and located closer from current pixel position. The proposed scheme handles the pixels in a conventional raster scan order.

Fig. 3a illustrates the current G pixel $g_{(i,j)}$ to predict and its candidate pixels, i.e., a set of 4 nearest G pixels that have been previously scanned. The predicted value of $g_{(i,j)}$, denoted as $\hat{g}_{(i,j)}$, is given by,

$$
\hat{g}_{(i,j)} = \sum_{(p,q)\in\zeta_1} w'_{(p,q)} \cdot g_{(p,q)}
$$

where $\zeta_1$ are 4 closest neighborhood pixels of $g_{(i,j)}$ such that $\zeta_1 \in \{(i-2,j-2),(i-2,j),(i-1,j),(i,j)\}$. The normalized weight factors, $w'_{(p,q)}$, are given by

$$
w'_{(p,q)} = w_{(p,q)} \sum_{(m,n)\in\zeta_1} w_{(m,n)}
$$

The original weight factor $w_{(p,q)}$ is defined as follows:

$$
w_{(p,q)} = \left[ 1 + \left( \sum_{(r,s)\in\zeta_1} \text{Diff}(T_{(p,q)}, T_{(r,s)})/D(G_{(p,q)}, G_{(r,s)}) \right) \right]^{-1}
$$

where $T_{(p,q)}$ is the template of G prediction centered at pixel $(p,q), T_{(p,q)} \in \{(p,q-2), (p-1,q-1), (p-2,q), (p-1,q+1)\}$, operator $\text{Diff}(\cdot)$ is a dissimilarity metric, and operator $D(\cdot)$ is a spatial distance between two pixels. We add 1 in the denominator to avoid a singularity issue that $\sum_{(r,s)\in\zeta_1} \text{Diff}(T_{(p,q)}, T_{(r,s)})/D(G_{(p,q)}, G_{(r,s)})$ becomes zero. The template used for G prediction is shown in Fig. 3b. Although using a larger template image in matching process improves prediction performance, the template of 4 pixels shows good trade-off between prediction accuracy and computational cost.

Typically, prediction techniques use sum of absolute differences (SAD) or sum square errors (SSE) between two templates in order to determine the degree of dissimilarity. We use the SAD due to its simplicity in implementation. Therefore, $\text{Diff}(T_{(p,q)}, T_{(r,s)})$ is defined as follows:

$$
\text{Diff}(T_{(p,q)}, T_{(r,s)}) = \\
|G_{(p-2,q)} - G_{(r-2,s)}| + |G_{(p-1,q-1)} - G_{(r-1,s-1)}| + \\
|G_{(p-2,q)} - G_{(r-2,s)}| + |G_{(p-1,q+1)} - G_{(r-1,s+1)}|
$$

As shown in Fig. 4, the proposed predictor requires a 5x7 support window centered at pixel location $(i-2, j-1)$ to calculate $\hat{g}_{(i,j)}$.

$$
w_{(i-2,j-2)}, w_{(i-2,j-1)}, w_{(i-2,j)}, w_{(i-1,j-1)}, \text{ and } w_{(i-1,j)} \text{, corresponding to the west, northwest, north, and northeast weight factors of } g_{(i,j)} \text{ pixel, are obtained using (7).}
Figure 3: (a) Current pixel to be predicted and its 4 closest neighborhood pixels in a quincunx G sub-image, (b) Template of G sub-image centered at (i,j). 'o' indicates pixels in the template region.

Figure 4: Pixel values required for the prediction of G pixel at (i,j).

\[
\begin{align*}
W_{g(i,j-2)} &= 1 + [(g(i,j-2) - g(i,j-4)) + |g(i-1,j-1) - g(i-1,j-3)| + |g(i-2,j-2)| + |g(i-1,j-1) - g(i-1,j-3)|](2)^{-1} \\
W_{g(i-1,j)} &= 1 + [(g(i,j-2) - g(i-1,j-3)) + |g(i-1,j-1) - g(i-2,j-2)| + |g(i-3,j-1)| + |g(i-1,j+1) - g(i-2,j)|](\sqrt{2})^{-1} \\
W_{g(i-2,j)} &= 1 + [(g(i,j-2) - g(i-2,j-2)) + |g(i-1,j-1) - g(i-3,j-1)| + |g(i-4,j)| + |g(i-1,j+1) - g(i-3,j+1)|](2)^{-1} \\
W_{g(i-1,j+1)} &= 1 + [(g(i,j-2) - g(i-1,j-1)) + |g(i-1,j-1) - g(i-2,j)| + |g(i-3,j+1)| + |g(i-1,j+1) - g(i-2,j+2)|](\sqrt{2})^{-1} \\
\end{align*}
\]

(7)

Once \( \hat{g}_{(i,j)} \) is obtained, G prediction error, \( e_{g(i,j)} \), is determined by \( e_{g(i,j)} = g(i,j) - \hat{g}_{(i,j)} \) and coded in the encoding module. Since the decoder can make same prediction \( \hat{g}_{(i,j)} \) as the encoder, the original G sub-image can be reconstructed without loss by adding decoded prediction error, \( e_{g}' \), and \( \hat{g}_{(i,j)} \).

3.3. Non-Green sub-image prediction

Independent encoding of deinterleaved sub-images yields suboptimal compression efficiency since data redundancy in the form of inter-channel correlation is disregarded during compression. In order to take into account inter-channel correlation, we perform the prediction of non-green sub-images in the color difference domain rather than the original intensity domain. To obtain color difference images, we need to estimate G samples at original R and B pixel locations, which are unavailable in original CFA layout. The missing G values are estimated from available G samples of the CFA image by interpolation. Various interpolation schemes are available from the low-complexity bilinear method to the complex methods utilizing a variety of estimation operators and edge-sensing mechanisms. Our simulation results have shown that advanced interpolation techniques typically improve the compression efficiency only marginally and thus, we use the simple bilinear approach.

Two color difference images, \( dr_{(i,j)} \) and \( db_{(i,j)} \) are defined as follows:

\[
\begin{align*}
\overline{d}_{r(i,j)} &= r_{(i,j)} - r_{(i,j)} , (i, j) \in \{(2m - 1, 2n - 1)\} \\
\overline{d}_{b(i,j)} &= b_{(i,j)} - b_{(i,j)} , (i, j) \in \{(2m, 2n)\} \\
\end{align*}
\]

(8)

where \( \overline{d} \) denotes interpolated G channels. Since prediction procedure of two color difference images, \( \overline{d}_{r(i,j)} \) and \( \overline{d}_{b(i,j)} \), are essentially identical, we only present a processing sequence for the red difference image using generalized difference signal \( d_{r(i,j)} \) in this section.
Similarly to G case, the proposed scheme predicts a current pixel \( d_{i,j}(p,q) \) using its four closest candidate pixels placed in the direction of west, northwest, north, and northeast, as shown in Fig. 5a. However, unlike G component, non-green components forms square lattices rather than quincunx ones, and hence, candidate pixels are defined to be \( c_2 \in \{(i-2,j), (i-2,j-2), (i-2,j), (i-2,j+2)\} \).

The prediction of color difference sub-images is also performed in a raster-scan order using the weighted template matching technique. The template for the color difference sub-image is defined in Fig. 5b using G samples, since edge and fine detail are typically deemphasized in color difference domain, while well preserved in G channel due to double sampling rate. The original weight factor of difference sub-image \( w_{d(p,q)} \) is defined as follows:

\[
\begin{align*}
 w_{d(p,q)} & = \\
 & = \left[ 1 + \left( \sum_{(r,s) \in c_2} \text{Diff}(T_{d(p,q)}, T_{d(r,s)})) / D(d_{p,q}, d_{r,s})) \right]^{-1}
\end{align*}
\]

where \( T_{d(p,q)} \) denotes the template of color difference image at \( (p,q) \), and defined as \( T_{d(p,q)} \in \{(p,q+1), (p,q-1), (p+1,q), (p-1,q)\} \).

\( w_{d(i,j-2)}, w_{d(i-1,j-1)}, w_{d(i-2,j)} \), and \( w_{d(i-1,j+1)} \), correspond to the west, northwest, north, and northeast weight factors of \( d_{i,j}(p,q) \) pixel, are obtained using (10).

\[
\begin{align*}
 w_{d(i,j-2)} & = \\
 & = \left[ 1 + (|l_{0}(i,j-1) - l_{0}(i,j-3)| + |l_{1}(i,j-1) - l_{1}(i,j-3)|) + |l_{0}(i,j+1) - l_{0}(i,j+3)| + |l_{1}(i,j+1) - l_{1}(i,j+3)|) / (2) \right]^{-1}
\end{align*}
\]

\[
\begin{align*}
 w_{d(i,j-2)} & = \\
 & = \left[ 1 + (|l_{0}(i,j-1) - l_{0}(i,j-3)| + |l_{0}(i,j-1) - l_{0}(i,j-3)|) + |l_{1}(i,j+1) - l_{1}(i,j+3)| + |l_{1}(i,j+1) - l_{1}(i,j+3)|) / (2 \sqrt{3}) \right]^{-1}
\end{align*}
\]

Once weight factors for all directions are computed, the predicted value is obtained using normalized weights \( w'_{d(p,q)} \) as follows:

\[
\hat{d}_{i,j} = \sum_{(p,q) \in c_2} (w'_{d(p,q)} \cdot d_{p,q})
\]

The prediction error of color difference images \( e_d \) is determined by \( e_d(i,j) = d_{i,j} - \hat{d}_{i,j} \) and coded in the encoding module. Again, the decoder has all information to make same prediction as the encoder and thus, it can reconstruct the R and B sub-image without loss.

3.4. Compression of prediction error

The prediction error signals of three sub-images, \( e_r \), \( e_g \), and \( e_b \), are obtained from previous stages. The distribution of prediction error signal is commonly modeled as Laplacian and Golomb-type coder is a widely used encoding solution to compress error signal due to its efficiency and simplicity [13, 16]. However we believe that maintaining compatibility with existing coding standard is a key element for large-scale adoption of the proposed compression scheme [27]. Therefore, in our study, we consider various existing image compression standards which supports lossless encoding of HDR input, such as JPEG-LS, JPEG 2000, and JPEG XR.

In the proposed scheme, we make use of JPEG XR due to the following reasons: i) JPEG XR supports channel bit-depth upto 24 bits for lossless compression, allowing efficient storage of HDR data, ii) JPEG XR yields balanced output between compression efficiency and computational complexity. JPEG XR typically provides almost comparable coding efficiency to high performance JPEG 2000 [28]. In terms of complexity, JPEG XR has considerably simpler architecture than JPEG 2000 and is comparable to low complexity JPEG-LS. Thus, we believe that JPEG XR is an ideal compression tool for resource constrained systems such as digital cameras, and having our scheme. In addition, use of JPEG XR will allow us to achieve near-lossless (perceptually lossless) encoding with minimal modification (e.g. by quantizing prediction errors of the residual image before entropy coding).

The number of samples to compress in \( e_g \) is twice as much as the ones in \( e_r \) and \( e_b \). It implies that the prediction error signal forms a standard 4:2:2 arrangement and thus, YCC 4:2:2 encoding mode of JPEG XR can be applied to compress it. To facilitate 4:2:2 encoding, the quincunx \( e_g \) plane is rearranged into a rectangular array by up-shifting every \( e_g \) pixels located in even rows by 1 pixel. \( e_r \) and \( e_b \) pixels are simply pressed together to form rectangular arrays.

Since the proposed prediction procedure significantly reduces the spatial correlation resides in the data, the block transform (i.e. LBT) prior to entropy coding within JPEG XR encoding pipeline is less effective in terms of data redundancy reduction. To address this issue, we modify JPEG XR codec software to facilitate
transform bypass mode that allows for direct encoding of residual samples without LBT and quantization process (Fig. 6). Such modification substantially reduces encoding complexity in the lossless coding, while maintaining compatibility with image coding standards. (Transform bypass mode is also employed in the intra-residual block lossless coding of the Fidelity Range Extensions high profiles of the H.264/AVC standard [29]) This bypass mode can be disabled if user want to reduce data redundancy in residual image for further bitrate reduction, or to yield near-lossless coding by quantizing prediction error signal.

4. Experimental Results

Experiments are carried out using 31 RGB images from the Para-Dice Insight Compression Database [30], shown in Figure 9. This database is chosen since it is a publicly available dataset containing a wide variety of RGB TIFF (Tagged Image File Format) images in 16-bit per component representation, varying in the edges and color appearances, and thus suitable for the evaluation of our proposed solution. Most images in the database are recorded in raw capture mode of high-end DSLR cameras (e.g. FUJI FinePix S5Pro, Leaf Aptus 22, etc) from various manufacturers, except for image 1 that is artificially generated using 3D modeling and ray-tracing. Three channel RGB images in the database are initially resized to 960x640 and sampled by the Bayer CFA to produce the grayscale CFA images \( z : Z^2 \rightarrow Z \). The CFA images \( z \) are then processed by the proposed pipeline and compressed into JPEG XR format \( c \) by JPEG XR reference software [31]. The reconstructed CFA images \( x : Z^2 \rightarrow Z \) are generated by applying JPEG XR decompression to the compressed data \( c \), followed by processing operations in decoding pipeline. As all intermediate steps are lossless, the reconstructed CFA images \( x \) should be identical to the original CFA images \( z \).

Performance of different solutions is evaluated by comparing lossless compression bitrate. Bitrate is reported in bits per pixel (bpp), \((8 \times B)/n\), where \( B \) is the file size in bytes of the compressed image including image header and \( n \) is the number of pixels in the image.

4.1. Primary color channel and color difference channel

This section compares the compression performance of original R/B channels and color difference channels. Fig. 7 shows the two-dimensional autocorrelation of the primary color images R and B, and the color difference images \( dr \) and \( db \), for the image 4 in our database. The height at each position indicates the correlation between the original image and spatially shifted version of itself, which is defined in (12):

\[
\text{Corr}(m, n) = \frac{\sum_i \sum_j (X(i,j) - \overline{X}(i,j))(X(i+m,j+n) - \overline{X}(i+m,j+n))}{\sqrt{\sum_i \sum_j (X(i,j) - \overline{X}(i,j))^2} \sqrt{\sum_i \sum_j (X(i+m,j+n) - \overline{X}(i+m,j+n))^2}}
\]

where \( X(i,j) \) is the original image, \( X(i+m,j+n) \) is the shifted version of itself, \( \overline{X} \) represent the mean values of the given image, and \( m, n \) denote spatial shifts in horizontal and vertical directions. The value at the center of graph is always 1 as it corresponds to zero shift case.

The figure shows that the level of similarity drops off more rapidly with color difference images than primary color images as shifting distance increases. This observation holds true for the other images in database. It
implies that \(dr\) and \(db\) have lower spatial correlation between neighborhood pixels than R and B. Since spatial redundancy is reduced by using color difference images, more efficient entropy coding is expected. As shown in Table 1, the proposed scheme yields average lossless compression bitrates of 12.563 bit per pixel (bpp) for primary color images and 11.974 bpp for color difference images, respectively.

4.2. Green channel interpolation method

Since we perform the weighted template matching prediction on the color difference domain, the estimation of missing G samples at R and B pixel positions is necessary. This is essentially achieved by interpolating the quincunx G image. In order to investigate the influence of an interpolation technique in coding performance, we examined several interpolation methods, including bilinear (BI), cubic spline interpolation (SPL), edge-directed interpolation (EDI) [32], new edge-directed interpolation (NEDI) [33], which vary in estimation accuracy and computational complexity. For BI, missing G samples are estimated by taking an average value of four surrounding pixels. In SPL, a piecewise continuous curve, passing through each of the given samples in G sub-image, is defined to determine missing pixel values. EDI is an adaptive approach that measures horizontal and vertical gradients of missing G samples to decide the direction to perform interpo-
Table 2: Lossless bitrate of proposed compression scheme with various G interpolation schemes

<table>
<thead>
<tr>
<th>Image</th>
<th>BI</th>
<th>SPL</th>
<th>EDI</th>
<th>NEDI</th>
<th>BI</th>
<th>SPL</th>
<th>EDI</th>
<th>NEDI</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>10.139</td>
<td>10.268</td>
<td>10.034</td>
<td>10.037</td>
<td>17</td>
<td>11.714</td>
<td>11.726</td>
<td>11.734</td>
</tr>
</tbody>
</table>

According to Table 3, the lossless bitrates for SAD and SSE are almost identical as 11.974 bpp and 11.975 bpp, respectively. We can conclude that selection of dissimilarity measure does not significantly affect compression performance and therefore, SAD is preferred to SSE due to its low complexity in implementation.

Table 3: Lossless bitrate of proposed compression scheme with SAD and SSE dissimilarity metrics

<table>
<thead>
<tr>
<th>Image</th>
<th>SAD</th>
<th>SSE</th>
<th>Image</th>
<th>SAD</th>
<th>SSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>10.139</td>
<td>10.161</td>
<td>17</td>
<td>11.714</td>
<td>11.716</td>
</tr>
<tr>
<td>2</td>
<td>13.028</td>
<td>13.032</td>
<td>18</td>
<td>13.147</td>
<td>13.155</td>
</tr>
<tr>
<td>3</td>
<td>13.021</td>
<td>13.027</td>
<td>19</td>
<td>13.147</td>
<td>13.155</td>
</tr>
<tr>
<td>4</td>
<td>11.015</td>
<td>11.015</td>
<td>20</td>
<td>11.307</td>
<td>11.308</td>
</tr>
<tr>
<td>5</td>
<td>11.717</td>
<td>11.716</td>
<td>21</td>
<td>12.365</td>
<td>12.369</td>
</tr>
<tr>
<td>6</td>
<td>10.239</td>
<td>10.246</td>
<td>22</td>
<td>11.888</td>
<td>11.887</td>
</tr>
<tr>
<td>7</td>
<td>10.249</td>
<td>10.258</td>
<td>23</td>
<td>10.792</td>
<td>10.791</td>
</tr>
<tr>
<td>8</td>
<td>10.787</td>
<td>10.784</td>
<td>24</td>
<td>11.292</td>
<td>11.293</td>
</tr>
<tr>
<td>9</td>
<td>12.920</td>
<td>12.922</td>
<td>25</td>
<td>11.662</td>
<td>11.654</td>
</tr>
<tr>
<td>10</td>
<td>13.317</td>
<td>13.314</td>
<td>26</td>
<td>12.300</td>
<td>12.299</td>
</tr>
<tr>
<td>12</td>
<td>11.146</td>
<td>11.148</td>
<td>28</td>
<td>12.191</td>
<td>12.189</td>
</tr>
<tr>
<td>13</td>
<td>12.889</td>
<td>12.888</td>
<td>29</td>
<td>10.507</td>
<td>10.492</td>
</tr>
<tr>
<td>16</td>
<td>10.662</td>
<td>10.670</td>
<td>Avg</td>
<td>11.974</td>
<td>11.975</td>
</tr>
</tbody>
</table>

4.3. Dissimilarity measure in template matching

The dissimilarity measure is a key element in template matching during prediction, since the choice of dissimilarity metric in (5) and (9) affects computational complexity and the accuracy of the prediction process. Table 3 presents the lossless compression bitrates of the proposed scheme for the images from our database using two commonly used dissimilarity metrics, SAD and SSE. They are defined as follows:

\[
SAD(i,j) = |i - j|
\]

\[
SSE(i,j) = (i - j)^2
\]

4.4. Comparative analysis with other lossless encoding schemes

based method combined with JPEG XR compression, vii) method 7: our proposed method, viii) method 8: YDgCoCg reversible color transform based approach [19] in junction with JPEG XR compression, ix) method 9: Mallat waveletpacket transform based approach [13].

As a basis for performance comparison, we apply some representative lossless compression schemes, such as JPEG XR, JPEG 2000, and JPEG-LS, directly on the CFA image in first three methods. Kakadu v.6.4 software implementation is used for JPEG 2000 coding and FFmpeg software is used for JPEG-LS coding.

Methods 4, 5, 6, 7 employ predictive coding techniques to facilitate lossless encoding of HDR CFA data. In method 4, quincunx G channel is separated into two rectangular lattices G1 and G2, and the prediction is carried out by estimating G1 from G2. Non-green channels are directly encoded in color difference domain. The CMBP predictor in method 5 makes use of context matching technique in prediction. For green channel, it initially generates a direction vector map of sample image to determine homogeneous/heterogeneous regions and only performs prediction in heterogeneous regions with pre-defined weight factors for neighborhood pixels. For non-green channel, prediction is carried out in color difference domain without such region classification process. The ALCM predictor in method 7 estimates a current pixel using a weighted combination of neighbor pixels. Initially equal weights are assigned for all pixels and if previous prediction was higher than the actual pixel value, then the weight of the largest neighbor pixel is decreased by 1/256 and the one for smallest neighbor pixel is increased by the same amount. If previous prediction was lower than the actual pixel value, then the weight of the largest neighbor pixel is increased by 1/256 and the one for smallest neighbor pixel is decreased by the same amount.

Methods 8 and 9 rely on different types of decorrelation technique to achieve lossless compression. Method 8 utilizes a fully reversible color transformation prior to applying JPEG XR image coding to decorrelate spectral redundancies in the CFA data. It maps original CFA mosaic image of $K_1 \times K_2$ into four rectangular images of $K_1/2 \times K_2/2$, each for one of the color channel, YDgCoCg. This new color space defines one luma channel Y that can be computed as an average of four original values in a 2x2 Bayer block, and three chrominance channels, Dg (difference green), Co(excess orange), and Cg(excess green). Then four sub-images are independently processed and combined to yield final encoded bitstream under JPEG XR encoding scheme. This method offers fully reversible integer based transform using lifting technique [34] and good spectral decorrelation properties with reduced complexity, requiring only simple addition and right-shift operators for implementation of it’s forward/inverse color transformation. Due to aforementioned strengths, in fact, Microsoft HD Photo codec software [20] includes this dedicated CFA encoding support within its implementation, providing the user with a mean of lossless coding of high bit-depth CFA image, and thus, we include this method as a meaningful benchmark algorithm for comparative analysis.

In method 9, the mosaic data is decorrelated by the 2D Mallat wavelet transform and the coefficients are then compressed by adaptive Rice code. Since four subbands of transform output contain low-frequency components of either chrominance or luminance signal, efficient energy packing is achieved with this method.

Fig. 8 shows the entropy of sample images from our database associated with different prediction schemes, from method 4 to 7. The entropy of image can be determined by the formula

$$H = - \sum_{i=1}^{n} P_i \log_2 P_i$$

where $P_i$ is probability of occurrence of pixel value $i$ and $H$ is the entropy of image. The entropy is evaluated by generating image histogram from the prediction error image of each sample images. Since the entropy of image data determines the theoretical lower bound which can be achieved by lossless compression, we can evaluate the effectiveness of different prediction algorithms.

The average entropies of various prediction methods result in 12.956, 11.637, 11.704, and 11.395 for method 4, 5, 6, and 7, respectively. The proposed method shows an improvement in 12.956, 11.637, 11.704, and 11.395 for method 4, 5, 6, and 7, respectively. The proposed method shows the lowest average entropy value, indicating potential high compression efficiency.

The output compression bitrates of CFA images from our database achieved by various methods are presented in Table 4. The results clearly show that direct compression of the CFA mosaic image is not efficient. In direct CFA compression scenario, JPEG 2000 is superior to JPEG XR and JPEG-LS in terms of compression efficiency, outperforming JPEG XR and JPEG-LS in average bitrate by 0.5 and 1.1 bpp, respectively. However, as it can be seen, higher compression efficiency can be achieved by exploiting redundancy reduction strategy in CFA data prior to encoding.

On average, our proposed scheme yields a lossless compression bitrate of 11.974 bpp for images in our database. The average compression bitrate obtained by other reviewed schemes (non direct CFA compression) are 13.240, 12.208, 12.233, 13.059, and 12.470 bpp, for
method 4, 5, 6, 8 and 9, respectively. In general, pixel by pixel prediction schemes (e.g. method 5, 6, and 7) outperforms other methods in comparison as they take into account both spectral and spatial relationship between samples. On the other hands, method 8 offers lower compression efficiency since it only reduces spectral redundancy in CFA data via color transformation. For most of images in database, the proposed method consistently achieves the lowest lossless compression bitrates, proving robustness of the solution in terms of compression efficiency.

Apart from the lossless bitrate performance of the proposed solution, its computational complexity is also analyzed in terms of normalized operations, such as addition (ADD), bit shift (SHF), multiplication (MUL), and absolute value (ABS). Table 5 presents a summary of number of operations per pixel required to carry out each stage of prediction process. In this analysis, the bilinear interpolation is used for missing G pixel estimation and SAD metric is used for dissimilarity measurement during prediction. It can be seen that performing non-green prediction in the color difference domain instead of the intensity domain increases number of operations for the proposed scheme by 2 addition and 0.5 shift per pixel since the G interpolation and the difference signal estimation stages are unnecessary for the intensity domain. Such a marginal increase in computational cost is considered to be tolerable given that use of the color difference domain yields reduction in average lossless bitrate by 0.5 bpp as shown in Section 4.1.

In summary, the following conclusion can be drawn:
i) the structure separation step reduces high frequency artifacts, leading to high compression efficiency, ii) the proposed weighted template matching predictor exploits inter-channel and spatial correlation to achieve high compression performance, and iii) the proposed scheme utilizes low complexity building blocks, such as bilinear interpolation, SAD dissimilarity measure, and JPEG XR encoding module with transform bypass mode, to minimize the computational cost. The image entropy analysis and experimentation conducted on sample HDR CFA images indicate that the proposed scheme delivers higher lossless compression performance than other prior-art solutions.

5. Conclusion

In this paper, we propose a novel lossless compression scheme to compress Bayer CFA images represented in HDR format. The proposed scheme deinterleaves the input CFA image into sub-images of single color component and adopts a predictor depending on local image statistics. Generated prediction error signals of each sub-image are then compressed by a JPEG XR encoding module, achieving HDR input support at low computational cost. Experimental result demonstrates that the proposed scheme effectively removes spatial and spectral redundancies, delivering superior compression efficiency than other prior-art solutions.
Table 4: Lossless bitrate of various CFA compression schemes

<table>
<thead>
<tr>
<th>Stage</th>
<th>ADD</th>
<th>SHF</th>
<th>MUL</th>
<th>ABS</th>
</tr>
</thead>
<tbody>
<tr>
<td>G sub-image prediction</td>
<td>19.5</td>
<td>1</td>
<td>7</td>
<td>8</td>
</tr>
<tr>
<td>G interpolation (BI)</td>
<td>1.5</td>
<td>0.5</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Diff R/B channel estimation</td>
<td>0.5</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Diff R sub-image prediction</td>
<td>9.75</td>
<td>0.5</td>
<td>3.5</td>
<td>4</td>
</tr>
<tr>
<td>Diff B sub-image prediction</td>
<td>9.75</td>
<td>0.5</td>
<td>3.5</td>
<td>4</td>
</tr>
<tr>
<td>Total</td>
<td>41</td>
<td>2.5</td>
<td>14</td>
<td>16</td>
</tr>
</tbody>
</table>

Table 5: Number of operations per pixel required for the proposed scheme

References

Figure 9: Test digital color images (referred to as image 1 to image 31, from left to right and top to bottom)