

An Enhanced Lossless Color Filter Array Image Compression Based on Predictive Adaptive Arithmetic Coding

L. M. Varalakshmi, R. Sowmiya

Department of Electronics & Communication Engineering, Sri Manakula Vinayagar Engineering College, India

Article Info

Article history:

Received Dec 11, 2014

Revised Feb 10, 2015

Accepted Feb 19, 2015

Keyword:

Color filter array

Context coding

Hierarchical prediction

Lossless compression

ABSTRACT

Most consumer digital cameras use a single image light sensor which provides color information using color filter array(CFA).This provided a mosaic images, in which each pixel position contains only one color component in case of Bayer CFA Pattern. This paper produced a CFA hierarchical prediction scheme based on context adaptive coding. In CFA hierarchical scheme, the green pixels were subdivided into two sets G_1 and G_2 . G_1 was encoded by a gray scale conventional method and G_2 was predicted based on G_1 . The red pixels were predicted using both the sets of green pixels and blue pixels were predicted using red and green. The predictors were designed based on direction of the edges in the neighborhood. Using the prediction information, the magnitude of prediction error was also determined and context adaptive arithmetic coding was applied to reduce bits. The simulated results on CFA images showed that the proposed method gives less bits per pixel than the recently developed CFA compression algorithms.

Copyright © 2015 Institute of Advanced Engineering and Science.
All rights reserved.

Corresponding Author:

L. M. Varalakshmi,

Department of Electronics & Communication Engineering,

Sri Manakula Vinayagar Engineering College,

Puducherry-605107, India.

Email: varalakshmi-1@yahoo.co.in

1. INTRODUCTION

With an increasing attractiveness of digital cameras, the quantity of digital images is considerably growing, and the resolution of digital images is also quickly increasing. Digital cameras are becoming gradually more popular in the user electronics market. To reduce the cost, most video cameras repeatedly capture the color information using a single charge-coupled device or complementary metal-oxide-semiconductor sensor imaging pipeline with the red-green-blue (RGB) color filter array (CFA) structure. The Bayer pattern CFA shown in Figure 1 is commonly used, which captures pixels which containing two green, one red and one blue sample (Figure 1).

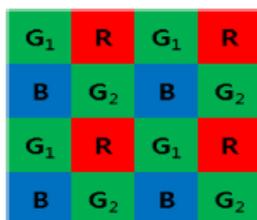


Figure 1. Bayer CFA pattern [1]

Color demosaicking (CDM) determines missing two color components in each pixel location of the CFA image to produce the full color image. In a conservative digital camera pipeline, CDM is initially performed on the CFA image, followed by compression of demosaicked image. A camera's image processor performs most of the preprocessing steps such as white balancing, denoising, and demosaicking using the raw data captured by CFA. The demosaicked color image is then saved as raw RGB image or sent to image encoder. Hence, by processing of these images in a system is sometimes redundant because the CFA data has been already processed using the camera's processor.

Data compression is another important processing stage in digital cameras, to reduce the storage level, redundancies are removed from the data. Compression techniques can be classified as either lossy or lossless. In lossless compression, the original image or data is retained exactly after decompression. Lossy compression achieves better compression ratios with some amount of distortion in the decompressed data.

To get the best compression performance, compressing CFA data is more efficient than compressing demosaicked color images, as represented in [4]. In particular, the compression scheme is more efficient than the demosaicking-first method because the demosaicking method increases the number of data points that are somewhat correlated. In the early lossy CFA compression methods mentioned above [5]–[7], the RGB components of CFA data are converted to decorrelated RGB components, which is independently encoded.

In this paper, we propose a new predictive coding scheme depends on hierarchical prediction method and a new context adaptive method which is enabled by the hierarchical scheme for RGB Images and CFA Images. For CFA Images, in hierarchical prediction, half of the G pixels are used for the prediction of the other half of G pixels, both sets of G pixels are used to predict R pixels, and Green and Red pixel values are then used to predict Blue pixels. For RGB Images, the components are first transformed or converted by a reversible color transform, and each of the transformed components are independently compressed by the above explained methods. Due to high density and highly correlated of RGB data and it cannot be compressed adeptly and it is changed to YCbCr by using reverse color transform. In this both CFA and RGB hierarchical scheme, already encoded pixels are used for context adaptive modeling, i.e. to find the conditional probability density function (pdf) of prediction error for the given neighboring pixels.

After the prediction and context adaptive modeling, the prediction errors also with the contexts are encoded using a conventional context-adaptive arithmetic encoder [16]. In the experiments, the proposed method is analyzed with recent predictive encoding method in [10], which says the best performance among the existing methods, and also with the modern transform method in [8]. Comparing the results on some simulated CFA data and also on real CFA data available in [20]. The comparison shows that the proposed method gives less bits per pixel on all of the images referenced above. The remaining part of this paper is organized as follows. In subdivision 2, the structure of our CFA encoder is presented. Then, Subdivision 3 deals with prediction scheme and Subdivision 4 presents the context modeling for adaptive encoding. Experiments on various simulated and real CFA and RGB data are explained in Subdivision 5, and the conclusion is presented in Subdivision 6.

2. OVERVIEW OF PROPOSED ENCODER

2.1. RGB Images

For the prediction of a pixel that is to be encoded, here we propose a method which makes use of lower row pixels as well as the upper and also left pixels. For the lossless compression of color components, the RGB is first altered to YC_uC_v by an RCTransform and Y-grey channel is determined by a conventional grayscale image compression algorithm. The signal dissimilarity is generally much smaller than that of RGB signal, but still bigger near the edges in the case of chrominance channels (C_u and C_v). For more defined prediction of these signals, and also for hierarchical modeling of prediction errors, we use the hierarchical scheme, the chroma image is splitted into two subimages; i.e. a pair of even numbered rows and a pair of odd numbered rows respectively. The even row subimage X_e is pre-arranged, then we can make use of all the pixels in X_e even no row for the prediction of a pixels in the odd row subimage X_o .

2.2. CFA Images

The proposed encoder's structure is shown in Figure 2, which has hierarchical predictor, a conventional grayscale encoder, and a context-adaptive arithmetic encoder. The input CFA image which is in Figure 1, the G pixels in the odd rows ($G1$ pixels) are encoded first using a conventional grayscale coder method. Next, they are used for the prediction of the G pixels in the even numbered rows ($G2$ pixels), which produces the prediction error that is represented as $eG2$. Third, the interpolating the G pixels, to fill in the G values at the positions of the Red and Blue pixels. Fourth, the interpolated G pixels are subtracted from the R and B pixels, producing the difference ΔR and ΔB . It is to note that ΔR and ΔB are used as an alternative of R and B, respectively, to exploit the channels correlation.

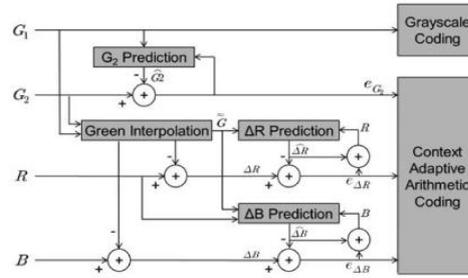


Figure 2. The proposed encoder's structure

Fifth, the ΔR values are predicted from the previously encoded neighboring ΔR values, along with the information obtained from the G pixels, and the prediction error which is represented as $e_{\Delta R}$ is obtained. At last, all of the previously encoded pixels (which includes the G, ΔR , and preceding ΔB pixels) are used to solve an appropriate predictor for a given ΔB , and the prediction error $e_{\Delta B}$ is generated. The prediction block obtains the error signals, i.e., e_{G2} , $e_{\Delta R}$, and $e_{\Delta B}$ are fed into the context-adaptive arithmetic encoder.

Algorithm (a): To find the direction of i, j .

```

if  $|x_o(i,j) - \hat{x}_h(i,j)| + T_1 < |x_o(i,j) - \hat{x}_v(i,j)|$  then
    dir(i,j) ← H
else
    dir(i,j) ← V
end if
    
```

Near the edges the large prediction errors are expected and within textured areas even if a much more elaborate predictor is used, which strictly degrades the performance of a conventional entropy method. However, if we are able to estimate the pdf of the error for the given neighbor pixels, more efficient encoding method is possible as mentioned in [18]. More precisely, when we encode an prediction error e_n at a pixel position n , we can make use of the information from the already encoded neighboring pixels as the context C_n . In new words, we can put up the pdf $P(e_n|C_n)$ while encoding, which is used for the adaptive arithmetic coding. Hence, if the context modeling is perfect and correct, even large errors are expected to be with the low entropy and thus the context adaptive encoder produces less bits than a normal entropy coder. The details of context modeling for the CFA data will be described in subdivision 4.

3. HIERARCHICAL PREDICTION OF CFA AND RGB DATA

3.1. Prediction for CFA Data

As shown in Figure 1, we denote a mosaic image as consists of four subimages: $G1$, $G2$, R , and B . As mentioned previously, we first encode $G1$ part by a grayscale compression method JPEG-Lossless, and then hierarchical scheme of prediction of other subimages follows.

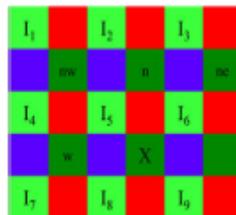


Figure 3. Illustration for the explanation of prediction of $G2$ pixel from the neighboring pixels. $G1$ pixels are denoted as light green, and the $G2$ as dark green

3.1.1. Prediction of G2

Let us reflect on the situation that a pixel denoted as x in Figure 3 is encoded.

Algorithm (b) To calculate the overall pixel prediction

```

if dir(i-1,j) = H or dir(i,j-1) = H then
    Calculate dir(i,j) by Algorithm (a)
    Program dir(i,j)
    if dir(i,j) = H then
         $\hat{x}_o(i,j) \leftarrow \hat{x}_h(i,j)$ 
    else
         $\hat{x}_o(i,j) \leftarrow \hat{x}_v(i,j)$ 
    end if
else
     $\hat{x}_o(i,j) \leftarrow \hat{x}_v(i,j)$ 
    Calculate dir(i,j) by Algorithm (a)
end if

```

Note that light green boxes are the pixels denoted in the odd rows ($G1$) and dark green boxes are represented in the even rows ($G2$). To predict x, we can use the already encoded pixels of $G2$ such as nw, n, ne, w, and all the neighboring pixels in $G1$. As in conventional prediction process, we define four varieties of directional predictors (horizontal diagonal, vertical, right diagonal, left diagonal) as in Equation (1).

$$\{P_h, P_v, P_{dr}, P_{dl}\} = \left\{w, n, \frac{I_5+I_9}{2}, \frac{I_6+I_8}{2}\right\} \quad (1)$$

Which consist of the adjacent pixels to the x, into the respective directions. In this proposed scheme, it is not needed to choose one out of these four predictors, but we.

To realize this design, we define a variable for the choose any two best predictors and combine them with appropriate weights. The choice of predictors and weights is of course depends on the direction of edges around the x. Edge directivity around x as represented as:

$$D_a = \frac{\sum(\text{Pixel difference around x into the direction a})}{\sum(\text{distances of pixel pairs})}$$

Where $a = h, v, dr, \text{ or } dl$.

3.1.2. Interpolation of Green Values in Positions of R and B

As mentioned above, the blue and red pixels are not encoded directly, but $\Delta R = R - \hat{G}$ and $\Delta B = B - \hat{G}$ are encoded directly and also predicted, where \hat{G} is a green interpolation value. Because, all the closest four neighbors are available in this case, we interpolate (predict) only into horizontal and vertical directions defined as in Equation (2).

$$\{P_h, P_v\} = \left\{\frac{I_6+I_7}{2}, \frac{I_4+I_9}{2}\right\} \quad (2)$$

3.1.3. Prediction of Red Pixels and Blue Pixels

Once obtaining the G values in the positions of the R and B, we can calculate the ΔR and ΔB . For predicting the ΔR in the position marked as x, the predictors are simply the neighboring ΔR s in the matching directions as in Equation (3).

$$\{P_h, P_v, P_{dr}, P_{dl}\} = \{w, n, nw, ne\} \quad (3)$$

3.2. Prediction Method for RGB Image

The chrominance (i.e) chroma channels Cu and Cv which results from the RCT usually have different labels from Y (luma), and also different from the original color planes R , G , and B in this hierarchical decomposition scheme. The overall signal dissimilarity is masked by the color transform in the chrominance channels, but the disparity is still large near the object boundaries. As a result, the prediction errors in a chroma channel are much reduced in a smooth (i.e) horizontal region, but remain reasonably great near the edge or within a texture region explained in Algorithm (a) and (b).

The pdf of prediction error for better context modeling, along with the precise prediction is estimated for the efficient compression. Here, we advise (propose) a hierarchical decomposition scheme that is pixels in an key(i:e) input image X is splitting into two subimages: an even subimage X_e and an odd subimage X_o . An even subimage is encoded first and is used to predict the pixels in odd subimage X_o .

In addition, X_e is also used to find the figures of prediction errors of X_o . For the compression of X_o pixels using X_e , directional prediction is worked to stay away from the large prediction errors close to the edges. For each pixel, the horizontal predictor $\hat{x}_h(i, j)$ and vertical predictor $\hat{x}_v(i, j)$ are defined as in Equation (4).

$$\hat{x}_v(i, j) = \text{round}\left(\frac{x_e(i, j) + x_e(i+1, j)}{2}\right), \quad (4)$$

And one among them is selected to predict $x_o(i, j)$. The most important one is the horizontal predictor because it will be more accurate only when there is a strong horizontal edges. To realize this design, we define a variable for the direction of edge at each pixel $\text{dir}(i, j)$, which is given either Horizontal or Vertical. To decide the direction of i, j and it is explained in Algorithm (a).

4. SIMULATION RESULTS

4.1. For RGB Images by RCT using Context Coding

On the various test images, this algorithm is applied, which is widely used for the lossless compression. In all the simulation results, the parameter T1 in Algorithm (b) and the quantity of contexts are assigned to 3 and 5. The conditional probability density function has been simulated as mentioned in Figure 4. The classic images which has a combination of Blue, Green and Red and it is denoted (changed) into YC_bC_r , then compression is performed and arithmetic coding has been applied to measure/calculate bit rate and PSNR.

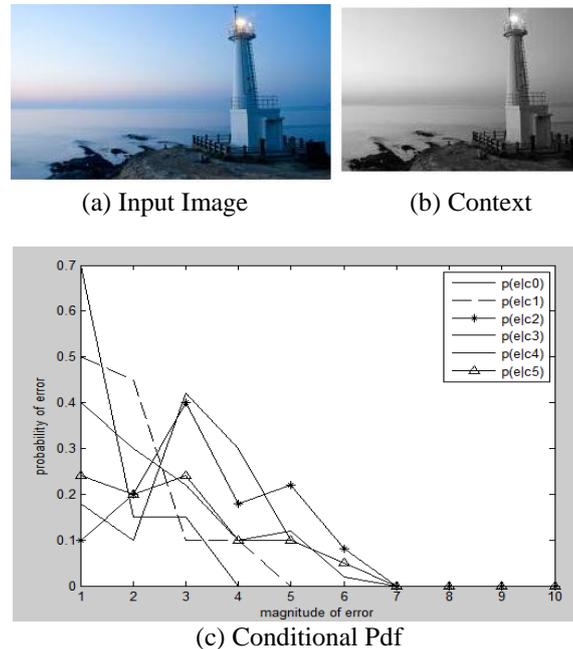


Figure 4. An example of local activity (context) and probability of error depending on context

The results which are simulated are summarized in Table 1, Table 2 and Table 3 which compares the compressed bit rates, PSNR and Coding time with existing methods.

Table 1. Comparison of Compressed Bit Rates (Bpp) For Classic Images

	Size	CALIC	JPEG XR	Proposed
Lena	512×512	13.1787	14.0942	13.5162
Peppers	512×512	13.8661	15.3245	09.0703
Eye	512×512	18.1511	18.2553	10.0641
Strawberry	512×512	14.9567	15.1408	12.8927

Table 2. Peak Signal to Noise Ratio (dB) For Classic Images

	Size	CALIC	Proposed
Lena	512×512	11.4589	13.5500
Peppers	512×512	07.7312	10.1209
Eye	512×512	07.1962	12.0532
Strawberry	512×512	06.6057	11.8927

Table 3. Comparison of CPU Times (Seconds) On A Pc Intel Core-I3-2.20 Ghz Cpu

Input Images	Entropy of Red Channel	Entropy of Green Channel	Entropy of Blue Channel	Joint Entropy
1	7.5535	7.612	7.6406	15.584
2	7.628	7.661	7.5789	15.449
3	7.0487	7.2962	7.1561	12.158
4	7.4264	7.4568	7.3406	14.585

4.1.1. Compressed Images

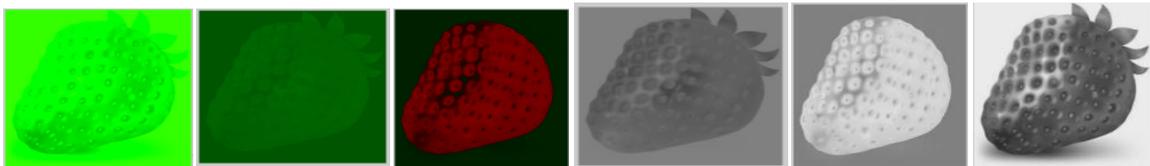


Figure 5. The Classic images set 1

4.2. For Color Filter Array Images

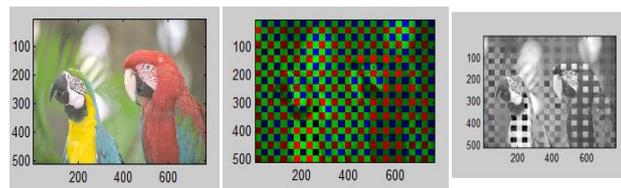


Figure 6. Cropped part of CFA image



Figure 7. Splitting of channels

a) Calculation of Compressed bit rates

Table 4 shows the compressed bit rate for each color filter array image where bpp represents the number of bits of information stored per pixel of an image.

Table 4. Calculation of Compressed bit rates

Images	Size	JPEG-XR	Proposed
1	768×512	8.8331	7.6462
2	768×512	8.8296	7.6806
3	768×512	8.1698	7.3385
4	768×512	7.9263	7.5623
Avg.		8.4397	7.5571

b) Calculation of Compression Ratio

Table 5 shows the compression ratio, it is ratio between the size of an original image to the size of the compressed image.

Table 5. Compression Ratio

Images	Size	Compression Ratio
1	768×512	1.9675
2	768×512	1.5841
3	768×512	1.3796
4	768×512	1.8613

c) Entropy of Channels

Table 6. Entropy of Channels

Images	Entropy of Red channel	Entropy of Blue Channel	Entropy of Green Channel	Joint Entropy
1	7.5535	7.6406	7.6120	15.584
2	7.6280	7.5789	7.6610	15.449
3	7.0487	7.1561	7.2962	12.158
4	7.4294	7.3406	7.4568	14.585

d) Measurement of PSNR and MSE

Table 7 represents the measurement of Mean square error and peak signal to noise ratio. It shows that higher the PSNR, lower will be mean square error.

Table 7. Measurement of PSNR and MSE

Images	Size	JPEG-XR		Proposed	
		MSE	PSNR	MSE	PSNR
1	768×512	1.7	37	1.1	40
2	768×512	1.8	36	1.6	38
3	768×512	1.4	35	1.3	37
4	768×512	2.2	33	1.8	35

5. CONCLUSION

A new lossless compression algorithm for the Bayer-patterned CFA images has been proposed. The proposed method predicts the color components in hierarchical manner based on context adaptive arithmetic coding. In hierarchical prediction, we encode the half of the green samples using a conventional grayscale encoder, and other half of the green samples are encoded based on first half of the encoded green samples. Red pixels are predicted using the green samples and blue pixels are predicted using red and green samples. For reducing the prediction residual. Edge directivity is considered. The proposed scheme is tested on simulated data that has been widely used and on the real CFA images and some additional high-resolution simulated CFA images. The results show that the proposed method provides reduced bits per pixel than the transform-based method and other existing methods.

REFERENCES

- [1] W. J. Yang, *et al.*, “Universal chroma subsampling strategy for compressing mosaic video sequences with arbitrary RGB color filter arrays in H.264/AVC”, *IEEE Trans. Circuits Syst. Video Technol.*, Vol. 23, No. 4, pp. 591-606, Apr 2013.
- [2] D. Lee and K. N. Plataniotis, “Lossless compression of HDR color filter array image for the digital camera pipeline”, *Signal Processing: Image Communication*, Vol. 27, No. 6, pp. 637-649, Jul 2012.
- [3] H. S. Malvar and G. J. Sullivan, “Progressive-to-lossless compression of color-filter-array images using macropixel spectral-spatial transformation”, In Proc. DCC, pp. 3-12, 2012.
- [4] K. H. Chung and Y. H. Chan, “A fast reversible compression algorithm for Bayer color filter array images”, In Proc. APSIPA, 2009, pp. 825-888.
- [5] C. Doutre, *et al.*, “H.264-based compression of Bayer pattern video sequences”, *IEEE Trans. Circuits Syst. Video Technol.*, Vol. 18, No. 6, pp. 725-734, Jun 2008.
- [6] K. H. Chung and Y. H. Chan, “A lossless compression scheme for Bayer color filter array images”, *IEEE Trans. Image Process.*, Vol. 17, No. 2, pp. 134-144, Feb 2008.
- [7] X. Lian, *et al.*, “Reversing demosaicking and compression in color filter array image processing: Performance analysis and modeling”, *IEEE Trans. Image Process.*, Vol. 15, No. 11, pp. 3261-3278, Nov 2006.
- [8] R. Lukac and K. N. Plataniotis, “Single-sensor camera image compression”, *IEEE Trans. Consum. Electron*, Vol. 52, No. 2, pp. 299-307, May 2006.
- [9] N. Zhang and X. L. Wu, “Lossless compression of color mosaic images”, *IEEE Trans. Image Process.*, Vol. 15, No. 6, pp. 1379-1388, Jun 2006.
- [10] N. X. Lian, *et al.*, “Reversing demosaicking and compression in color filter array image processing: Performance analysis and modeling”, *IEEE Trans. Image Process.*, Vol. 15, No. 11, pp. 3261-3278, Nov 2006.
- [11] B. K. Gunturk, *et al.*, “Demosaicking: Color filter array interpolation”, *IEEE Signal Process. Mag.*, Vol. 22, No. 1, pp. 44-54, Jan 2005.
- [12] C. C. Koh, *et al.*, “New efficient methods of image compression in digital cameras with color filter array”, *IEEE Trans. Consum. Electron*, Vol. 49, No. 4, pp. 1448-1456, Nov 2003.
- [13] B. K. Gunturk, *et al.*, “Color plane interpolation using alternating projections”, *IEEE Trans. Image Process.*, Vol. 11, No. 9, pp. 997-1013, Sep 2002.
- [14] S. Y. Lee and A. Ortega, “A novel approach of image compression in digital cameras with a Bayer color filter array”, In Proc. IEEE Int. Conf. Image Process., pp. 482-485, Oct. 2001.
- [15] Information Technology–JPEG 2000 Image Coding System–Part 1: Core Coding System, INCITS/ISO/IEC Standard 15444-1, 2000.
- [16] X. Wu and N. Memon, “Context-based, adaptive, lossless image coding”, *IEEE Trans. Commun.*, Vol. 45, No. 4, pp. 437-444, Apr 1997.
- [17] B. E. Bayer, “Color imaging array”, *U.S. Patent 3 971 065*, Jul 1976.