A Study on Lossless Compression of Color Image and Color Filter Array Image

Ph.D. Dissertation

Seyun Kim

Department of Electrical Engineering and Computer Science

Seoul National University

ABSTRACT

Lossless image compression is less used than lossy compression due to its large memory or bandwidth requirements. However in some fields, such as medical, prepress, scientific, and artistic areas, lossy compression cannot substitute for lossless compression. As cameras and display systems are going high quality and as the cost of memory is lowered, we may also wish to keep our precious and artistic photos free from compression artifacts. Hence efficient lossless compression will become more and more important, although the lossy compressed images are usually satisfactory in most cases. In this dissertation, algorithms using interchannel correlation are studied, with which lossless compression schemes for color image and color filter array image are proposed.

At first, a new reversible color transform (RCT) is proposed, which consists of the conventional RCT and additional lifting steps to further decorrelate chroma images Cu and Cv. Red, green, and blue samples in an image are highly correlated, but YC_bC_r transform shows good decorrelation performance for RGB images. However because the transform cannot be used for lossless compression, RCT is used for lossless coding standards including JPEG2000. Due to its invertibility and simplicity, the decorrelation performance of the conventional RCT is not satisfying, so the improvement is required. With effective but simple operations, the proposed scheme

shows much higher decorrelation performance than the conventional RCT, and the improved performance is comparable with YC_bC_r. In addition, lossless bit rates of JPEG-LS, the standard lossless image coder, after color transforms are presented, in which the proposed RCT outperforms the conventional RCT over 1.46% with minimum increase in operations.

Next, a lossless color image compression method based on a new hierarchical encoding scheme is proposed. Specifically, an input RGB image is transformed into $YC_u{}'C_v{}'$ color space using the new RCT. After the color transformation, the luminance channel Y is compressed by a conventional lossless image coder. The chrominance channels are encoded with the proposed hierarchical decomposition and directional prediction. Finally, an appropriate context modeling of prediction residuals is introduced and generic arithmetic coding is applied. The proposed method and several conventional methods are tested on the Kodak image set, some medical images, and digital camera images, and it is shown that average file size reductions over JPEG2000 for these sets are 5.85%, 10.40%, and 4.89% respectively. When the mode selection is tried, further encoding gain can be obtained.

At last, a new lossless compression method for Bayer color filter array (CFA) images is proposed, which focuses on efficient context modeling. For the efficient modeling of prediction errors, hierarchical prediction scheme is adopted, in which input mosaic image is divided into four subimages, and the subimages are encoded in order. For the prediction of a subimage, all of subimages which are already encoded are used to estimate edge direction and candidate predictors. The already encoded subimages and pixels in causal neighborhood are also used to estimate the magnitude of prediction error, and the prediction error is encoded by adaptive arithmetic coder along with the estimated context. The proposed method is test for real CFA images

and simulated CFA images from Kodak set and commercial digital camera images, and it outperforms all the compared methods.

Key words: lossless color image compression, reversible color transform, lifting, hierarchical coding, color filter array compression

Student number: 2007-20944

Contents

Abstract **i**

Contents **iv**

List of Figures **vii**

List of Tables **x**

1 Introduction **1**

1.1 Lossless Color Image Compression 1

1.2 Interchannel Correlation 2

1.3 Lossless Compression of CFA Image 3

1.4 Outline of This Dissertation 5

2 Related Works **9**

2.1 Lossless Image Compression Using Interchannel Correlation 9

2.1.1 Interband CALIC 9

2.1.2 RCT 12

2.2 Lossless Compression of Mosaic Image 13

2.2.1 LCMI 13

2.2.2 CMBP 16

3 New Reversible Color Transform **21**

3.1 Introduction 21

3.2 Proposed Method 22

3.3 Experimental Results 23

3.4 Conclusion 25

4 Hierarchical Prediction Scheme for Lossless Color Image Compression **29**

4.1 Introduction 29

4.2 Proposed Method 30

4.2.1 Hierarchical Decomposition 30

4.2.2 Directional Prediction 31

4.2.3 Proposed Coding Scheme 32

4.3 Experimental Results 34

4.4 Conclusion 38

5 Color Filter Array Compression **45**

5.1 Introduction 45

5.2 Overview of Proposed Encoder 49

5.3 Hierarchical Prediction of CFA data 50

5.3.1 Prediction of G_2 51

5.3.2 Interpolation of green values in the positions of R and B . . . 54

5.3.3 Prediction of red and blue pixels 55

5.4 Encoding Prediction Errors 57

5.5 Experimental Results 59
5.5.1 The Proposed Method 59
5.5.2 Demosaic-first and Compression-first Schemes 62
5.6 Conclusion 66

6 Conclusions **71**

Bibliography **75**

Abstract (Korean) **83**

List of Figures

1.1 CIE Color Matching Functions (from [8]). 6

1.2 Red, green, and blue channels of Lena image. 6

1.3 $YCbCr$ transform result of Lena image. 7

1.4 Bayer CFA pattern [12]. 7

2.1 The labeling of neighboring pixels used for prediction and modeling in interband CALIC. 10

2.2 Lena image and signals in object boundary (white line). 18

2.3 Lighthouse (Kodak 19) image and part of its mosaic image. 18

2.4 Decomposed subbands of mosaic image of Lighthouse (Kodak 19). . 19

2.5 The green channel compression of CMBP. 19

3.1 Flowgraph of the proposed RCT. 22

3.2 The Kodak images. 27

4.1 Input image and its decomposition. 40

4.2 Illustration of one level decomposition. 40

4.3 Context modeling (Kodak19). 40

4.4 The medical images. 41

4.5 The digital camera images. 42
4.6 The Classic images. 43

5.1 Bayer CFA pattern [12]. 46
5.2 Structure of the proposed encoder. 50
5.3 Illustrations for the explanation of (a) prediction of G_2 pixel from the neighboring pixels, and (b) interpolation of missing green pixel (at the position of R and B). 51
5.4 (a) Prediction of dR pixels, and (b) prediction of dB pixels. 56
5.5 Entropy of resiual with respect to δ. 58
5.6 The Kodak images. 67
5.7 The real CFA images (demosaicked). 68
5.8 The NIKON D90 images. 68
5.9 The OLYMPUS E-P1 images. 69
5.10 Rate-distortion performances of lossy color image compression (demosaic-first) and lossy CFA compression (compression-first). The x-axis denotes bit per pixel (bpp) and the y-axis denotes CPSNR. 70

List of Tables

3.1 Inter-channel correlation for 24 Kodak images. 24

3.2 Lossless bit rates of JPEG-LS for 24 Kodak chroma images. 25

4.1 Lossless bit rates of various schemes (bpp) for Kodak images. 36

4.2 Lossless bit rates for the medical images. 37

4.3 Lossless bit rates for the commercial digital camera images. 37

4.4 Lossless bit rates of various schemes (bpp) for Classic images. 38

4.5 Summary of experimental results. 38

5.1 Comparison of the algorithms for a set of real CFA images. 60

5.2 Comparison of the algorithms for a set of simulated CFA images from KODAK set. 61

5.3 Comparison of the algorithms for a set of simulated CFA images from digital cameras. 62

5.4 Lossless bit rates of various schemes (bpp) for the simulated CFA images from OLYMPUS E-P1 set (4032×3024 resolution). 63

5.5 Lossless bit rates of demosaic-first and compression-first schemes (bpp) for the simulated CFA images (Kodak). 64

5.6 Lossy bit rates and CPSNRs of demosaic-first and compression-first schemes for the simulated CFA images (Kodak). 65

Chapter 1

Introduction

1.1 Lossless Color Image Compression

Digital images are usually encoded by lossy compression methods due to their large memory or bandwidth requirements. The lossy compression methods achieve high compression rate at the cost of image quality degradation. However, there are many applications where the loss of information or artifact due to compression needs to be avoided, such as medical, prepress, scientific and artistic images. For example, white spot indicating cancer in magnetic resonance imaging (MRI) might be removed by lossy compression. As cameras and display systems are going high quality and as the cost of memory is lowered, we may also wish to keep our precious and artistic photos free from compression artifacts. Hence efficient lossless compression will become more and more important, although the lossy compressed images are usually satisfactory in most cases.

Along with the standardization or independently, many lossless image compression algorithms have been proposed. Among a variety of algorithms, the most widely

used ones may be Lossless JPEG [1], JPEG-LS [2], LOCO-I [3] and CALIC [4]. The LOCO-I and CALIC were developed in the process of JPEG standardization, where most ideas in LOCO-I are accepted for the JPEG-LS standard although the CALIC provides better compression performance at the cost of more computations.

Above mentioned methods are developed for the grayscale images, which can of course be applied to each channel of hyperspectral or color images independently. For achieving higher coding gain than the independent compression of each channel, inter-channel correlation is used in [5, 6]. Specifically, the Interband CALIC [5] assumes linear relationship between the reference channel and a channel to be encoded. However, the actual relationship is more complicated than just being a linear model. Focusing on the RGB color images (rather than hyperspectral images), there is another approach to reducing the channel correlation, based on the color transform. For example, the RGB to YC_bC_r transform may be the most frequently used one for the lossy compression of color image and video. However, in the case of lossless compression, most color transforms cannot be used due to their uninvertibility with integer arithmetic. Hence an invertible version of color transform, the reversible color transform (RCT) was defined and used in JPEG2000 [7]. For the color image compression, the RCT followed by intraband compression usually provides higher coding gain than the independent compression.

1.2 Interchannel Correlation

A pixel in most digital color images is represented as three values–red, green, and blue. It is because human visual system has three types of cone cells, whose spectral response functions are investigated by Commission Internationale de L′Éclairage

(CIE) and the result is illustrated in Fig. 1.1 [8]. Because the response functions are widely overlapped with each other, there exists high interchannel correlation between red, green, and blue channel. For example, light ray with wavelength of 600nm affects both type of primaries, $\bar{x}$ and $\bar{y}$, to make the primaries highly correlated. During image capturing process, image in XYZ color space is transformed to RGB color space, which is proper to most display devices. After the transformation, interchannel correlation still exists in the RGB color space, and the example for Lena image is shown in Fig. 1.2.

To eliminate the correlation and improve compression efficiency, color transform is used in most image and video compression method such as JPEG [9], MPEG [10], and H.264 [11]. The RGB to YC_bC_r transform is most frequently used one for lossy compression. The transform is mathematically derived from the response functions of XYZ primaries and their transformation into RGB color space, and generates much more decorrelated three channels (Fig. 1.2). It can be observed that chroma images, C_b and C_r, are much smoother and carry less information than luma image Y. Quantitatively, correlation coefficient between red and green channels is 0.8786 and that between blue and green channels is 0.9106. After the transformation, correlation coefficients are sharply decreased into 0.7408 and 0.1209.

1.3 Lossless Compression of CFA Image

Due to advantages in cost, size, and power consumption, most digital cameras use CFA to make color images with single chip sensor. With CFA, only a single color component is measured at each pixel position, and the resultant CFA image is called mosaic image because pixels from different color components are interleaved

as shown in Fig. 5.1, which is the most popular Bayer pattern [12]. Through demosaicking [13] [14], an interpolating process, the other missing color components are estimated to make full resolution color image. Conventionally, CFA image is first demosaicked, and the full resolution image is compressed for storage or transmission with lossy or lossless scheme.

Recently, however, it is reported that the conventional demosaicking-first scheme is not efficient because demosaicking produces additional redundancy into mosaic image [15]. To solve this problem, various compression-first schemes, demosaicking after compression/decompression, are proposed, and they allow offline high performance post-processing which cannot be carried out in portable devices due to heavy computational cost.

It is typical lossy compression that widely used for general purpose photography because image quality and compression performance of it is sufficient to nonprofessional users. In addition, redundancy introduced by demosaicking is reduced with lossy compression more effectively than with lossless compression. However, there are many applications where any information loss or compression artifact is not permitted, such as medical, professional printing, scientific and artistic images. For lossless compression, compression rate is more important problem than lossy coding, and this is why lossless CFA compression has been used by commercial digital cameras including ones from Sony.

Some lossless CFA methods [16] [17] including many of lossy CFA compression methods [18] [19] [20] use spectral-spatial transform to make typical grayscale subimages from mosaic image, which is followed by an existing image coder such as JPEG2000 [7] and JPEG-LS [2]. In [16], Mallat wavelet packet transform is shown to be beneficial for CFA image compression, and [17] proposed spectral-spatial trans-

form for macropixel of CFA pattern. However, the strategy of transform to typical subimages and standard coding has a limit that spectral correlation between transformed subimages is not used after the transformation. Therefore, methods which have dedicated prediction structure for CFA image show better compression performance thanks to increased system complexity [21] [22] [23], which first encode green pixels and use the green pixels to predict red and blue pixels.

1.4 Outline of This Dissertation

The rest of this dissertation is organized as follows. First, conventional techniques for lossless image compression are reviewed in Chapter 2, and the proposed RCT for lossless color image compression is presented in Chapter 3. In Chapter 4, we explain the hierarchical decomposition, detailed prediction process, and the proposed coding scheme for lossless color image coding. In Chapter 5, lossless CFA compression scheme is proposed. Finally, Section 6 concludes the dissertation.

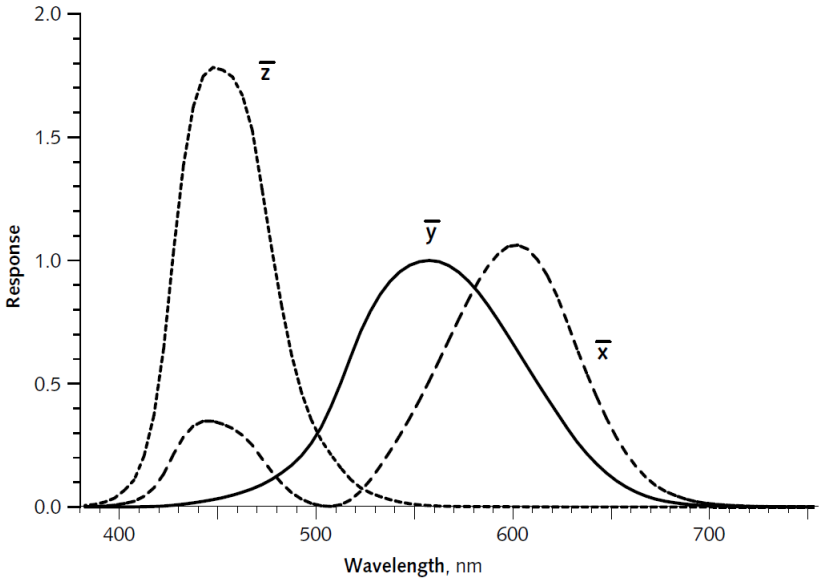


Figure 1.1: CIE Color Matching Functions (from [8]).

(a) Red channel (b) Green channel (c) Blue channel

Figure 1.2: Red, green, and blue channels of Lena image.

(a) Y channel (b) Cb channel (c) Cr channel

Figure 1.3: $YCbCr$ transform result of Lena image.

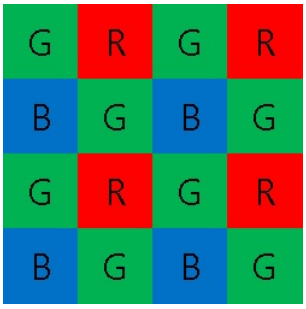


Figure 1.4: Bayer CFA pattern [12].

Chapter 2

Related Works

In this chapter, conventional techniques for lossless image compression are reviewed, which are related to the proposed methods. First, the Interband CALIC [5] is explained to understand existing interchannel prediction method used for losselss color image compression. In addition, the conventional RCT used for JPEG2000 is described. Next, recent compression-first scheme, or CFA compression, is compared to the conventional demosaic-first scheme, and two existing methods, LCMI and CMBP, are explained.

2.1 Lossless Image Compression Using Interchannel Correlation

2.1.1 Interband CALIC

Interband version of CALIC [4] is presented in [5], which uses linear model to predict a pixel in current frame from pixels in reference frame. In this method, every pixel x_i in reference band X is encoded with intraband CALIC, and then a pixel y in

current band Y is predicted using x and casual neighborhood pixels of y and x, which are illustrated Fig. 2.1. At first, the correlation coefficient between current band Y and reference band X is measured as

$$\rho(X,Y) = \frac{8\sum_{i=1}^{8} x_i y_i - \sum_{i=1}^{8} x_i \sum_{i=1}^{8} y_i}{\sqrt{[8\sum_{i=1}^{8} {x_i}^2 - (\sum_{i=1}^{8} x_i)^2][8\sum_{i=1}^{8} {y_i}^2 - (\sum_{i=1}^{8} y_i)^2]}}. \tag{2.1}$$

When $\rho(X,Y)$ is high, or larger than threshold T_ρ, it can be assumed that strong similarity exists between the two bands in the neighborhood of current pixel. In this case, linear model $\hat{Y} = \alpha X + \beta$ is used to predict y. The parameters which minimize $||\hat{Y} - Y||_2$ are known to be

$$\alpha = \frac{8\sum_{i=1}^{8} x_i y_i - \sum_{i=1}^{8} x_i \sum_{i=1}^{8} y_i}{8\sum_{i=1}^{8} {x_i}^2 - (\sum_{i=1}^{8} x_i)^2} \tag{2.2}$$

and

$$\beta = \frac{8\sum_{i=1}^{8} y_i - \alpha \sum_{i=1}^{8} x_i}{8}. \tag{2.3}$$

When $\rho(X,Y)$ is sufficiently high, $\hat{y} = \alpha x + \beta$ might be a good predictor for y. If a sharp edge is near y, however, the least-square fitting from the eight neighboring pixels may not be accurate at the location of y. To avoid this problem, the approximations

	x_7	x_6	
x_8	x_3	x_2	x_4
x_5	x_1	x	

(a) Reference Band

	y_7	y_6	
y_8	y_3	y_2	y_4
y_5	y_1	y	

(b) Current Band

Figure 2.1: The labeling of neighboring pixels used for prediction and modeling in interband CALIC.

$$y - y_1 = \hat{y} - \hat{y_1} = \alpha(x - x_1) \quad \text{and} \quad y - y_2 = \hat{y} - \hat{y_2} = \alpha(x - x_2) \tag{2.4}$$

are used to get the predictors

$$\hat{y_h} = y_1 + \alpha(x - x_1) \quad \text{and} \quad \hat{y_v} = y_2 + \alpha(x - x_2), \tag{2.5}$$

where $\hat{y_h}$ and $\hat{y_v}$ are extrapolations along with horizontal and vertical directions. $\hat{y_h}$ is more accurate than $\hat{y_v}$ when horizontal gradient $|x - x_1|$ is smaller than vertical gradient $x - x_2$ and vice versa. Therefore, the following gradient-adjusted predictor is used.

$$\hat{y} = \begin{cases} \hat{y_h} & \text{if } |x - x_2| - |x - x_1| > T \\ \hat{y_v} & \text{if } |x - x_2| - |x - x_1| < -T \\ (\hat{y_h} + \hat{y_v})/2 & \text{otherwise.} \end{cases} \tag{2.6}$$

This interframe predictor outperforms intraframe predictors if $\rho(X, Y)$ is high. When the correlation is low, however, it is better to use intraframe predictors. Therefore, if $\rho(X, Y) > T_\rho$ then the interframe predictor is used; otherwise intraframe predictor defined in CALIC is used.

In this method, a pixel in current frame is predicted by just one reference frame, but most color images have three or more channels. Method using color transforms can simultaneously decorrelate all channels in an image although the decorrelation performance might be unsatisfying. In addition, interchannel correlation is modeled as linear relationship in interband CALIC, but first order model is not sufficient especially near object edges, which can be shown in Fig. 2.2. In the method,

moreover, pixels are processed in raster scan order, and prediction is carried out by extrapolation. However, this extrapolation often fails when object boundary or sharp edge exist near current pixel location because ratio between different channel signals changes near the region.

2.1.2 RCT

The RGB components of images are highly correlated with each other, so decorrelating color transforms are used before encoding each component. For example, the YC_bC_r transform widely used for lossy compression of image and video is defined as

$$\begin{bmatrix} Y \\ Cb \\ Cr \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ -0.16875 & -0.33126 & 0.5 \\ 0.5 & -0.41869 & -0.08131 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \tag{2.7}$$

where the elements of the matrix are derived from spectral responses of red, green, and blue color filters of standard RGB system. But, this cannot be used for lossless compression because there is no exact inverse transform required for manipulating integer pixel values. Hence, in the case of JPEG2000 lossless scheme, a reversible color transform is introduced as

$$\begin{aligned} Y &= \left\lfloor \frac{R+2G+B}{4} \right\rfloor & & G = Y - \left\lfloor \frac{C_u + C_v}{4} \right\rfloor \\ C_u &= R - G & \Longleftrightarrow \quad & R = C_u + G \\ C_v &= B - G & & B = C_v + G \end{aligned} \quad . \tag{2.8}$$

This is named "reversible" because the transform is exactly invertible with integer arithmetic. Ignoring effect of the floor operations, this transform can be expressed as

$$\begin{bmatrix} Y \\ C_u \\ C_v \end{bmatrix} = \begin{bmatrix} 1/4 & 1/2 & 1/4 \\ 1 & -1 & 0 \\ 0 & -1 & 1 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \Longleftrightarrow \begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} 1 & 3/4 & -1/4 \\ 1 & -1/4 & -1/4 \\ 1 & -1/4 & 3/4 \end{bmatrix} \begin{bmatrix} Y \\ C_u \\ C_v \end{bmatrix}. \tag{2.9}$$

Since this transform is focused on simplicity and invertibility with integer arithmetic, it does not provide satisfying decorrelation performance. Hence, a new RCT that can better decorrelate the channels is proposed in Chapter 3.

2.2 Lossless Compression of Mosaic Image

2.2.1 LCMI

To make CFA image, most digital cameras use Bayer pattern [12] as shown in Fig. 5.1. Letting $R(y,x)$, $G(y,x)$, and $B(y,x)$ be respectively red, green, and blue samples at pixel position (y,x), CFA image $I(y,x)$ is defined as

$$I(y,x) = \begin{cases} R(y,x) & \text{if } y \text{ is even and } x \text{ is odd} \\ B(y,x) & \text{if } y \text{ is odd and } x \text{ is even} \\ G(y,x) & \text{otherwise.} \end{cases} \tag{2.10}$$

Part of CFA image might be shown as

$$\begin{matrix} G_{0,0} & R_{0,1} & G_{0,2} & R_{0,3} & G_{0,4} & R_{0,5} \\ B_{1,0} & G_{1,1} & B_{1,2} & G_{1,3} & B_{1,4} & G_{1,5} \\ G_{2,0} & R_{2,1} & G_{2,2} & R_{2,3} & G_{2,4} & R_{2,5} \\ B_{3,0} & G_{3,1} & B_{3,2} & G_{3,3} & B_{3,4} & G_{3,5} \\ G_{4,0} & R_{4,1} & G_{4,2} & R_{4,3} & G_{4,4} & R_{4,5} \\ B_{5,0} & G_{5,1} & B_{5,2} & G_{5,3} & B_{5,4} & G_{5,5} \end{matrix}, \tag{2.11}$$

where red, green, and blue samples are interleaved as shown in Fig. 2.3(b), and spatial correlation in the mosaic image is difficult to be used without addition processing.

To make mosaic image into more suitable form in the image compression aspect, discrete wavelet transform (DWT) is used to deinterleave the red, green, and blue samples. The low-pass and high-pass filters of biorthogonal 5/3 CDF wavelet transform are

$$f_L = \left(-\frac{1}{8}, \frac{2}{8}, \frac{6}{8}, \frac{2}{8}, -\frac{1}{8}\right) \quad \text{and} \quad f_H = \left(-\frac{1}{2}, 1, -\frac{1}{2}\right). \tag{2.12}$$

Applying f_H horizontally and vertically in turn, 2-D 5/3 HH filter is derived as

$$F_{HH} = f_H^T f_H = \frac{1}{4}\begin{bmatrix} 1 & -2 & 1 \\ -2 & 4 & -2 \\ 1 & -2 & 1 \end{bmatrix}. \tag{2.13}$$

This filter is only used for $G_{2m+1,2n+1}$ because decimation is performed in DWT process. At $G_{3,3}$ position, DWT output $J_{3,3}$ is

$$J_{3,3} = G_{3,3} - \frac{2}{4}(R_{2,3} + R_{4,3} + B_{3,2} + B_{3,4}) + \frac{1}{4}(G_{2,2} + G_{2,4} + G_{4,2} + G_{4,4}), \quad (2.14)$$

where the green pixels are low-pass filtered and the result is subtracted by the red and blue pixels. The subtraction makes an effect of interchannel prediction. With the same manner, the LH, HL, and LL filters are defined as

$$F_{LH} = f_H^T f_L = \frac{1}{16}\begin{bmatrix} 1 & -2 & -6 & -2 & 1 \\ -2 & 4 & 12 & 4 & -2 \\ 1 & -2 & -6 & -2 & 1 \end{bmatrix}, \quad (2.15)$$

$$F_{HL} = f_L^T f_H = \frac{1}{16}\begin{bmatrix} 1 & -2 & 1 \\ -2 & 4 & -2 \\ -6 & 12 & -6 \\ -2 & 4 & -2 \\ 1 & -2 & 1 \end{bmatrix}, \quad (2.16)$$

and

$$F_{LL} = f_L^T f_L = \frac{1}{64}\begin{bmatrix} 1 & -2 & -6 & -2 & 1 \\ -2 & 4 & 12 & 4 & -2 \\ -6 & 12 & 36 & 12 & -6 \\ -2 & 4 & 12 & 4 & -2 \\ 1 & -2 & -6 & -2 & 1 \end{bmatrix}. \quad (2.17)$$

Like the HH filter, the other three filters operate as a low-pass filter in intraband and simultaneously as a predictor in interband. Decomposed LL, LH, HL, and HH

subbands are defined as

$$
\begin{aligned}
LL(y,x) &= J_{2y,2x} = F_{LL} * I_{2y,2x} \\
LH(y,x) &= J_{2y+1,2x} = F_{LH} * I_{2y+1,2x} \\
HL(y,x) &= J_{2y,2x+1} = F_{HL} * I_{2y,2x+1} \\
&\text{and} \\
HH(y,x) &= J_{2y+1,2x+1} = F_{HH} * I_{2y+1,2x+1}
\end{aligned}
\tag{2.18}
$$

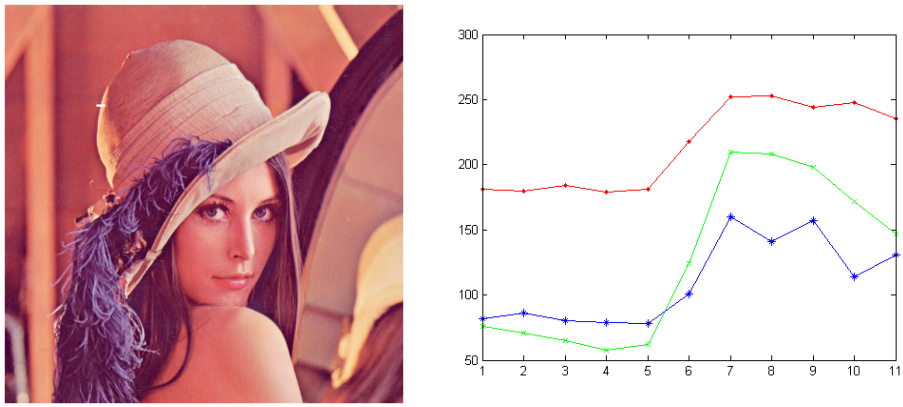
where $*$ denotes 2D convolution. An example of the decomposition is illustrated in Fig. 2.4, where each subband is deinterleaved so that it has homogeneous samples in spatial and spectral aspects unlike the interleaved mosaic image in Fig. 2.3(b). In LCMI, the decomposed subbands are independently encoded with JPEG2000 (Lossless mode).

2.2.2 CMBP

Among many lossless CFA compression methods [16,17,21–23], CMBP [21] provides very high compression rates. This method compresses green pixels first, and uses the encoded green pixels to predict and encode the rest red and blue pixels. In this process, the channel correlation is utilized in the form of color difference coding. To be precise, the green pixels are compressed in raster scan order. They are predicted from the information of already encoded pixels, and the prediction errors are encoded. For example, the prediction $\hat{g}$ for the pixel in Fig. 2.5 is a weighted sum of the candidate pixels C_1, C_2, C_3, and C_4 as

$$
\hat{g} = \sum_{k=1}^{4} w_i C_i, \tag{2.19}
$$

where w_i are the weights, and the four candidate pixels among the neighboring green pixels are determined by a context matching technique. In the case of red and blue pixels, the green pixels at the corresponding positions are first estimated, and then the color difference images are also compressed by predictive coding. The prediction residuals are modified to have exponential distribution and the Rice code is applied.



(a) Lena image (b) Red, green, and blue signals

Figure 2.2: Lena image and signals in object boundary (white line).

(a) Lighthouse (Kodak 19) image (b) Part of the mosaic image

Figure 2.3: Lighthouse (Kodak 19) image and part of its mosaic image.

Figure 2.4: Decomposed subbands of mosaic image of Lighthouse (Kodak 19).

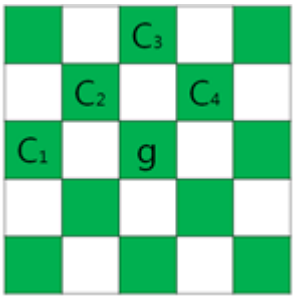


Figure 2.5: The green channel compression of CMBP.

Chapter 3

New Reversible Color Transform

3.1 Introduction

Because the components of RGB images are highly correlated with each other, a decorrelating color transform is used before encoding each component. For example, the YC_bC_r transform for lossy compression is defined as

$$\begin{bmatrix} Y \\ Cb \\ Cr \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ -0.16875 & -0.33126 & 0.5 \\ 0.5 & -0.41869 & -0.08131 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \tag{3.1}$$

where the elements of the matrix are derived from spectral responses of red, green, and blue color filters of standard RGB system. But, this cannot be used for lossless compression because there is no exact inverse transform required for manipulating integer pixel values. Hence, in the case of JPEG2000 lossless scheme, a reversible

color transform is introduced as

$$\begin{aligned} Y &= \left\lfloor \frac{R+2G+B}{4} \right\rfloor & G &= Y - \left\lfloor \frac{C_u + C_v}{4} \right\rfloor \\ C_u &= R - G & \Longleftrightarrow \quad R &= C_u + G \\ C_v &= B - G & B &= C_v + G \end{aligned} \quad . \tag{3.2}$$

This is named "reversible" because the transform is exactly invertible with integer arithmetic. Since this transform is focused on simplicity and invertibility with integer arithmetic, it does not provide satisfying decorrelation performance. Hence, a new RCT that can better decorrelate the channels is proposed.

3.2 Proposed Method

For the improvement of decorrelation performance, we modify the computations for C_v and C_u, which will lead the transform matrix more close to the YC_bC_r computation. In this process, we need to keep the additional computations as small as possible, and of course need to keep the invertibility. In the conventional RCT in (3.2), it can be seen that B is not involved in the computation of C_u and R is missing in C_v, whereas they are actually needed as shown in (3.1). For including the missing components, the best method may be to add the lifting process followed by the original RCT as shown in Fig. 3.1. This system gives the output

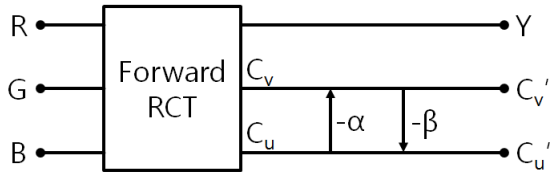


Figure 3.1: Flowgraph of the proposed RCT.

$$\begin{aligned} C_v' &= C_v - \lfloor \alpha C_u \rfloor \\ C_u' &= C_u - \lfloor \beta C_v' \rfloor \end{aligned} \quad \Longleftrightarrow \quad \begin{aligned} C_u &= C_u' + \lfloor \beta C_v' \rfloor \\ C_v &= C_v' + \lfloor \alpha C_u \rfloor \end{aligned} \tag{3.3}$$

where α and β are the coefficients that will lead the result close to YC_bC_r. The floor operations on $\lfloor \alpha C_u \rfloor$ and $\lfloor \beta C_v' \rfloor$ are required for the existence of inverse transform with integer arithmetic. Note that C_u' is calculated with C_v' instead of C_v. Another expression of (3.3) without the floor function is

$$\begin{bmatrix} C_v' \\ C_u' \end{bmatrix} \approx \begin{bmatrix} -\alpha & -1+\alpha & 1 \\ 1+\alpha\beta & -1+\beta-\alpha\beta & -\beta \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \tag{3.4}$$

which shows that all the color components are now involved for the computation of chrominance channel. Comparing (3.4) and (3.1), $\alpha \approx 2 \times 0.16875$ and $\beta \approx 2 \times 0.08131$ may be reasonable choices. For the implementation with only shift and addition, we approximated them as $\alpha = 1/4$ and $\beta = 1/8$.

3.3 Experimental Results

Table 3.1 shows the magnitudes of correlation coefficients of RGB channels for a set of Kodak images shown in Fig. 3.2. The table also shows the channel correlations when they are transformed by the conventional and proposed RCT. It can be shown that the chrominance images of the proposed RCT shows lower correlation to the luminance image Y, compared to those of the existing RCT. It also shows that the correlation by proposed RCT is close to the correlation by YC_bC_r.

Table 3.1: Inter-channel correlation for 24 Kodak images.

	RGB		YC_bC_r		RCT		Proposed RCT	
	R/G	B/G	C_b/Y	C_r/Y	U/Y	V/Y	U'/Y	V'/Y
1	0.8627	0.9914	-0.5207	-0.0856	-0.5686	-0.6767	-0.5241	-0.0882
2	0.5279	0.9740	0.0591	-0.2038	-0.0231	-0.3393	0.0442	-0.1879
3	0.7185	0.5534	-0.0925	-0.3883	-0.2287	-0.3214	-0.1037	-0.3657
4	0.6004	0.9568	-0.4609	0.4880	-0.4602	0.1385	-0.4638	0.4961
5	0.8981	0.9034	-0.1219	-0.3854	-0.2049	-0.3985	-0.1255	-0.3778
6	0.9772	0.9911	0.4413	-0.8502	0.3216	-0.8749	0.4384	-0.8485
7	0.8330	0.9084	0.0449	-0.0720	0.0103	-0.0015	0.0391	-0.0559
8	0.9665	0.9746	-0.0596	-0.4779	-0.1781	-0.5699	-0.0650	-0.4730
9	0.9486	0.8560	0.1209	-0.4896	-0.0479	-0.4753	0.1125	-0.4796
10	0.9555	0.9695	-0.0056	-0.5907	-0.1432	-0.6835	-0.0105	-0.5841
11	0.8225	0.9744	-0.3167	-0.0166	-0.3621	-0.2761	-0.3204	-0.0131
12	0.9133	0.9674	-0.5092	-0.1005	-0.5592	-0.3713	-0.5125	-0.0936
13	0.9818	0.9639	-0.1830	0.0207	-0.2007	0.0134	-0.1858	0.0282
14	0.8613	0.7107	0.1138	-0.5458	-0.0628	-0.5183	0.1021	-0.5320
15	0.8602	0.9880	-0.5175	-0.0802	-0.5552	-0.5933	-0.5210	-0.0774
16	0.9842	0.9511	0.0376	-0.3919	-0.1272	-0.4190	0.0309	-0.3797
17	0.9852	0.9779	-0.3027	-0.1784	-0.4315	-0.2931	-0.3071	-0.1806
18	0.9163	0.8362	0.4836	-0.6084	0.3929	-0.4883	0.4780	-0.5978
19	0.9672	0.9158	0.3025	-0.2824	0.2460	-0.2010	0.2918	-0.2722
20	0.9955	0.9764	-0.1116	0.0873	-0.1383	0.1140	-0.1209	0.0908
21	0.8957	0.9155	-0.2371	0.0018	-0.3535	-0.0161	-0.2508	0.0159
22	0.8714	0.8752	0.0595	0.0573	0.0576	0.1507	0.0553	0.0707
23	0.6147	0.6408	0.0095	-0.2343	-0.0582	-0.1277	0.0013	-0.2109
24	0.9761	0.9649	-0.1048	-0.1577	-0.1586	-0.1687	-0.1123	-0.1551
Avg.	**0.8722**	**0.9057**	**0.2174**	**0.2831**	**0.2454**	**0.3429**	**0.2174**	**0.2781**

Table 3.2 shows lossless bit rates of JPEG-LS for chroma images of Kodak set. Using proposed RCT before JPEG-LS, bit rates for U is slightly increased (1.27%) but bit rates for V is rather decreased (4.02%). The tendency is similar to that of the inter-channel correlation results, in which gain of proposed RCT for V is larger than that for U. Using the proposed RCT before JPEG-LS gives 1.46% of overall gain over the conventional RCT.

Table 3.2: Lossless bit rates of JPEG-LS for 24 Kodak chroma images.

	RCT		Proposed RCT	
	U	V	U'	V'
1	2.5963	2.6436	2.5874	2.4987
2	2.9253	2.4661	2.9362	2.4334
3	2.1637	2.4193	2.1736	2.3010
4	2.9383	2.4573	2.9441	2.2813
5	2.9833	3.1825	2.9846	3.0619
6	2.3805	2.7760	2.4688	2.6812
7	2.3638	2.6259	2.3792	2.4827
8	2.9495	3.0797	2.9649	2.9979
9	2.5112	2.6758	2.5255	2.4880
10	2.5377	2.7238	2.5497	2.5739
11	2.5427	2.3975	2.5575	2.2865
12	2.2841	2.4867	2.2902	2.3262
13	2.9150	3.4007	2.9301	3.2938
14	2.7361	2.8393	2.7692	2.8035
15	2.7770	2.6115	2.7668	2.4715
16	2.1848	2.4725	2.2053	2.3565
17	2.4487	2.6263	2.4538	2.4969
18	3.1241	3.3010	3.1307	3.2121
19	2.6889	2.8229	2.7039	2.7221
20	1.7982	2.9350	2.2520	2.9033
21	2.5643	2.9483	2.6076	2.8724
22	3.0289	3.1538	3.0244	3.0572
23	2.6317	2.7300	2.6377	2.6004
24	2.7160	2.9704	2.7439	2.8593
Avg.	**2.6163**	**2.7811**	**2.6495**	**2.6692**

3.4 Conclusion

In this Chapter, a new reversible color transform is proposed, which consists of the conventional RCT and additional lifting steps to decorrelate chroma images Cu and Cv further. Nearly optimal but simple parameters are found, and the

proposed scheme shows comparable decorrelation performance with YC_bC_r which can be applied only to lossy compression. In addition, lossless bit rates of standard lossless images coder are presented, in which the proposed RCT outperforms the conventional RCT over 1.46% with minimum increase in operation.

Figure 3.2: The Kodak images.

Chapter 4

Hierarchical Prediction Scheme for Lossless Color Image Compression

4.1 Introduction

In this Chapter, a new hierarchical prediction scheme is developed. While most of existing prediction methods in lossless compression are based on simple raster scan prediction, it is sometimes inefficient in the high frequency region. To be specific, we propose a prediction scheme that can use lower row pixels as well as the upper and left pixels. An RGB image is first transformed to YC_uC_v by the RCT, and Y channel is encoded by a conventional grayscale image compression algorithm. In the case of chrominance channels (C_u and C_v), the signal variation is generally smaller than that of RGB, but still large near the edges. For more accurate prediction of these signals, and also for accurate modeling of prediction errors, we use the a

hierarchical coding scheme. To be precise, the chrominance image is decomposed into two subimages; i.e. a set of even numbered rows and a set of odd numbered rows respectively. Once the even row subimage X_e is encoded, we can use all the pixels in X_e for the prediction of a pixel in the odd row subimage X_o. This means we can use the pixels in the lower rows as well as the upper and left pixels for the prediction of a pixel, which sometimes results in more accurate prediction compared to the conventional raster scan method. In addition, since the statistical properties of two subimages are not much different, the pdf of prediction errors of a subimage can be accurately modeled from the other one, which contributes to better context model for arithmetic coding. A subimage can be further decomposed, but we try only a one-level decomposition where the X_e is once more decomposed into even and odd column subimages.

4.2 Proposed Method

4.2.1 Hierarchical Decomposition

The chrominance channels $C_u{}'$ and $C_v{}'$ resulting from the RCT usually have different statistics from Y, and also different from the original color planes R, G, and B. In the chrominance channels, overall signal variation is suppressed through the color transform, but the variation is still large near the object boundaries. Hence, prediction errors in the chrominance channels are much reduced in smooth region, but remain relatively large near the edge or texture region.

For the efficient lossless compression, it is important to accurately estimate the pdf of prediction error for better context modeling, along with the accurate prediction. For this, we propose a hierarchical decomposition scheme as depicted in Fig.

4.1, which shows that pixels in an input image X is separated into two subimages: an even subimage X_e and an odd subimage X_o. Then, X_e is encoded first and is used to predict the pixels in X_o. In addition, X_e is also used to estimate the statistics of prediction errors of X_o. This decomposition and prediction can be carried out in a multi-level manner.

4.2.2 Directional Prediction

For the compression of X_o pixels, directional prediction is employed to avoid large prediction errors near the edges. For each pixel $x_o(i,j)$ in X_o, horizontal predictor $\hat{x_h}(i,j)$ and vertical predictor $\hat{x_v}(i,j)$ are defined as

$$\begin{aligned} \hat{x_h}(i,j) &= x_o(i,j-1) \\ \hat{x_v}(i,j) &= round\left(\frac{x_e(i,j)+x_e(i+1,j)}{2}\right), \end{aligned} \tag{4.1}$$

and one of them is selected as a predictor for $x_o(i,j)$. With these two possible predictors, the most common approach to encoding is "mode selection," where better predictor for each pixel is selected and the mode (horizontal or vertical) is also transmitted as side information. However, the vertical predictor is more often correct than the horizontal one when the predictors are defined as (4.1) because upper and lower pixels are used for the "vertical" whereas just a left pixel is used for the "horizontal". The horizontal predictor is more accurate only when there is a strong horizontal edge. Hence, vertical predictor is used for most pixels, and mode selection is used only when the pixel seems to be on a strong horizontal edge.

For implementing this idea, we define a variable for the direction of edge at each pixel $dir(i,j)$, which is given either H or V. Actually, it is given H only when the horizontal edge is strong, and given V for the rest. Deciding $dir(i,j)$ is

Algorithm 1 Calculation of $dir(i,j)$.

if $|x_o(i,j) - \hat{x_h}(i,j)| + T_1 < |x_o(i,j) - \hat{x_v}(i,j)|$ **then**
  $dir(i,j) \leftarrow H$
else
  $dir(i,j) \leftarrow V$
end if

Algorithm 2 Calculation of $\hat{x_o}(i,j)$.

if $dir(i-1,j) = H$ or $dir(i,j-1) = H$ **then**
  Calculate $dir(i,j)$ by Algorithm 1
  Encode $dir(i,j)$
  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 1
end if

summarized in Algorithm 1, where it can be seen that the direction is given H only when $|x_o(i,j) - \hat{x}_h(i,j)|$ is much smaller than $|x_o(i,j) - \hat{x_v}(i,j)|$ by adding a constant T_1 to the former when comparing them.

Based on the directions of pixels, the overall prediction scheme is summarized in Algorithm 2. It can be seen that the mode selection is tried when more than one of $dir(i-1,j)$ or $dir(i,j-1)$ are H, and the vertical prediction is performed for the rest.

4.2.3 Proposed Coding Scheme

In this section, we explain the overall process of image compression, including the new encoding scheme. An input RGB color image is transformed into $YC_u{}'C_v{}'$ color space by the proposed RCT. The luminance image Y is encoded by any of

lossless grayscale image coders, such as CALIC, JPEG-LS, or JPEG2000 lossless. The chrominance images C_u' and C_v' are encoded using the method described in Section 3. To be specific, a chrominance image $X^{(0)} \in \{C_u', C_v'\}$ is decomposed row by row into an even subimage $X_e^{(1)}$ and an odd subimage $X_o^{(1)}$ as shown in Fig. 4.2. The subimage $X_o^{(1)}$ is predicted and encoded using $X_e^{(1)}$, as described in Section 3. The subimage $X_e^{(1)}$ can be further decomposed column by column into the even subimage $X_e^{(2)}$ and the odd subimage $X_o^{(2)}$ as shown in the last figure of Fig. 4.2, where the subimage $X_o^{(2)}$ is compressed using $X_e^{(1)}$. For this "one level" decomposition image, the $X_e^{(2)}$ can be encoded by a standard image coder, or we can further decompose $X_e^{(2)}$ (multi-level decomposition) and process the encoding hierarchically.

In the predictive lossless compression, efficient encoding of the prediction error $e(i,j) = x_o(i,j) - \hat{x_o}(i,j)$ plays an important role. Although the proposed prediction method usually generates small prediction errors owing to new RCT and the sophisticated prediction scheme, there are still relatively large errors near the edge or texture region, which degrades the compression performance. For the efficient compression, the statistics of symbols (prediction errors) should well be described by an appropriate model and/or parameters. We model the prediction error as a random variable with pdf $P(e|C_n)$, where C_n is coding context that reflects the magnitude of edges and textures. Specifically, C_n is the level of quantization steps of pixel activity $\sigma(i,j)$ defined as

$$\sigma(i,j) = |x_e(i,j) - x_e(i+1,j)|. \tag{4.2}$$

Note that the local activity and its quantization steps are calculated with the pixels

in X_e, because all the pixels of X_e are available and its statistical property would be almost the same as that of X_o. The local activity is quantized into K steps such that C_n represents the step

$$q_{n-1} \leq \sigma(i,j) < q_n \tag{4.3}$$

for $n = 1, \cdots, K$ with $q_0 = 0$ and $q_K = \infty$. The length of quantization steps is determined such that each step includes the same number of elements (local activities). For each context, a generic adaptive arithmetic coder [24] is used to encode the prediction error. For illustration, Fig. 4.2.3 shows an input image, the local activity of a subimage (context), and $P(e|C_n)$ for several C_n. It describes the statistical property of prediction error very well, in that the error magnitude is large when the local activity is strong. Hence the proposed model can be effective for the compression with arithmetic coding.

4.3 Experimental Results

As stated in the introduction, the state-of-the-art lossless compression method may be the CALIC [4], which shows higher coding gain than the JPEG-LS (or LOCO-I) [2] [3] at the cost of higher computational complexity. For the compression of color image, the M-CALIC and JPEG2000 lossless provide better coding efficiency than the independent coding of each channel by CALIC or CALIC followed by RCT. Hence we compare the proposed method with the CALIC, CALIC after RCT, M-CALIC, and JPEG2000 lossless.

We first apply the algorithm on Kodak image set [28], which is widely used for the test of lossless compression [29–31] and demosaicking [32]. In all the experi-

ments, the parameter T_1 in Algorithm 2 and number of contexts K are set to 3 and 6 respectively, and each input image is one-level decomposed. The luminance images and decomposed highest level images $X_e^{(2)}$ in Fig. 4.2 are encoded by JPEG2000 lossless. Experiments are summarized in Table 5.4, which shows that the proposed method performs better for most images and the maximum gain over JPEG2000 lossless is 9.01% for the image 9. On the average, the proposed algorithm improves 5.85%, 14.55%, and 40.74% over JPEG2000 lossless, M-CALIC, and CALIC respectively. In Table 5.4, the "proposed (RCT)" means that the proposed hierarchical encoding is applied to the conventional RCTed images. From the last two columns, new RCT and the new encoding scheme both contribute to the coding gain for these images.

The proposed method is also tested on some of medical images in Fig. 4.4 and compared with JPEG2000 in Table 4.2. The result for M-CALIC is not shown here because the code provided by the authors does not work for the image size over 1024x1024. When only some part of images are compressed for this reason, the coding gain shows similar trend as Table 5.4. The test medical images are positron emission tomography (PET) images for human brain, digital camera images for eyes and eyeground, and endoscope images for human intestine, which are generally smooth and hence less bits are generated when compared with the case of Kodak images. On the average, the proposed algorithm produced 10.40% less bits than JPEG2000 lossless.

In addition, experiments for images from commercial digital cameras (shown in Fig. 4.5) are also conducted, and the results are compared in Table 4.3. The first five images are captured with NIKON D90, and the rest are captured with OLYMPUS E-P1. On the average, the proposed algorithm produces 4.89% less bits

Table 4.1: Lossless bit rates of various schemes (bpp) for Kodak images.

	CALIC	M-CALIC	JPEG2K	Proposed (RCT)	Proposed
1	18.0224	11.2800	10.3844	9.8765	**9.6803**
2	13.7745	10.2870	9.1628	8.7501	**8.6156**
3	12.5939	8.8260	8.0917	7.5406	**7.4545**
4	14.2316	10.2120	9.1116	8.6669	**8.4811**
5	17.7021	12.0720	10.8167	10.4720	**10.2921**
6	16.9876	10.4970	9.5911	9.2385	**9.1535**
7	13.5490	9.6060	8.5039	8.0021	**7.8692**
8	17.9877	12.4230	11.1389	10.7668	**10.6165**
9	13.0903	9.7260	8.9045	8.2458	**8.1020**
10	13.5027	9.7680	9.0564	8.4700	**8.3057**
11	15.7682	10.4940	9.2918	8.8744	**8.6958**
12	13.8552	9.2640	8.6577	8.1067	**7.9601**
13	19.6935	13.0740	11.8608	11.5393	**11.3734**
14	16.9641	11.4240	10.1605	9.6818	**9.5768**
15	12.9272	9.8040	8.9967	8.6241	**8.4379**
16	15.3833	9.6360	8.7748	8.1853	**8.0705**
17	13.9580	9.8670	9.0644	8.4054	**8.2557**
18	16.7664	12.3480	10.7706	10.5167	**10.3532**
19	15.1774	10.7850	9.6655	9.1798	**9.0475**
20	11.7572	**7.6200**	8.0769	8.1534	8.5058
21	16.4392	10.9020	9.7621	9.3302	**9.2145**
22	15.3030	11.4150	10.0939	9.8458	**9.7116**
23	12.0347	9.4650	8.5047	8.0375	**7.8719**
24	15.7731	11.0940	10.1673	9.7110	**9.5975**
Avg.	15.1351	10.4954	9.5254	9.0925	8.9684

than JPEG2000 lossless.

Finally, it is worth to note that the proposed method does not always perform best for every set of images. Specifically, the proposed hierarchical encoding scheme sometimes works better and sometimes worse than the conventional methods, depending on image sets and also depending on the channels (Y, $C_u{}'$, and $C_v{}'$). It is also true for every compression algorithms, i.e. the coding gain of compression algorithms differ on different set of images. For example, on the set of classical test

Table 4.2: Lossless bit rates for the medical images.

	JPEG2K	Proposed
PET1	6.7390	**6.0314**
PET2	7.3403	**6.5233**
PET3	7.0232	**6.3074**
Eye1	5.7498	**4.6807**
Eye2	5.4467	**4.4916**
Eyeground	3.2763	**3.0460**
Endoscope1	7.3532	**7.0701**
Endoscope2	5.1304	**4.9087**
Avg.	6.0074	5.3824

Table 4.3: Lossless bit rates for the commercial digital camera images.

	JPEG2K	Proposed
Ceiling	7.5571	**7.2423**
Locks	7.4574	**7.1785**
Flamingo	7.0366	**6.6535**
Berry	7.2468	**6.9479**
Sunset	6.3586	**6.0411**
Flower	6.4141	**6.1307**
Park	5.8977	**5.5945**
Fireworks	5.7797	**5.3289**
Avg.	6.7185	6.3897

images such as Lena, Peppers, and Mandrill, even the channel independent CALIC sometimes performs better than JPEG2000 and our algorithm, as shown in Table 4.4. Considering this variability on different set and different channels, we may combine the existing methods and our hierarchical encoding scheme for enhancing the compression gain of color images. More precisely, note that our algorithm can employ any of grayscale image compression methods (including the proposed hierarchical scheme) for the compression of Y and lowest level images in $C_v{}'$ and $C_u{}'$. Like H.264 [11] and HEVC [33] which heavily use "mode selection," where many methods and parameters are tried and the best one is selected, we may also try

Table 4.4: Lossless bit rates of various schemes (bpp) for Classic images.

	CALIC	M-CALIC	JPEG2K	Proposed	Proposed with mode selection
Lena	**13.1787**	13.2570	13.5848	13.7690	13.2354
Peppers	**13.8661**	13.9560	14.8000	15.2072	14.7024
Mandrill	18.1551	**17.7990**	18.0939	18.5813	18.1291
Barbara	14.9567	12.1500	**11.1612**	11.6044	11.6044
Avg.	15.0392	14.2905	14.4100	14.7905	14.4178
Bike	15.4654	N/A	**11.8153**	12.0074	12.0074
Cafe	17.9384	N/A	**14.0001**	14.2420	14.1299
Woman	14.9120	N/A	**11.2629**	11.3883	11.3883
Avg.	16.1053	N/A	12.3594	12.5459	12.5085

Table 4.5: Summary of experimental results.

	JPEG2K	Proposed	Proposed with mode selection
Kodak	9.5254	8.9684	**8.9473**
Medical	6.0074	5.3824	**5.1215**
Digital	6.7185	6.3897	**6.3012**
Classic 1	**14.4100**	14.7905	14.4178
Classic 2	**12.3594**	12.5459	12.5085

different grayscale compression algorithms for Y and highest level images in $C_v{}'$ and $C_u{}'$, and choose the best one along with just several bits of side information. The last column in Table 4.4 is the proposed method with this mode selection scheme (mode selection only for Y is tried), which shows that the proposed color compression scheme achieves better coding gain. Table 4.5 also shows that further coding gain is achieved by the mode selection scheme, for each of image sets tested above.

4.4 Conclusion

We have proposed a lossless color image compression method based on a new reversible color transform that provides higher decorrelation performance, and also a

new hierarchical encoding scheme. Specifically, an input RGB image is transformed into $YC_u{}'C_v{}'$ color space using the new RCT. After the color transformation, the luminance channel Y is compressed by a conventional lossless image coder. The chrominance channels are encoded with the proposed hierarchical decomposition and directional prediction. Finally, an appropriate context modeling of prediction residuals is introduced and generic arithmetic coding is applied. The proposed method and several conventional methods are tested on the Kodak image set, some medical images, and digital camera images, and it is shown that average file size reductions over JPEG2000 for these sets are 5.85%, 10.40%, and 4.89% respectively. When the mode selection is tried, further encoding gain can be obtained.

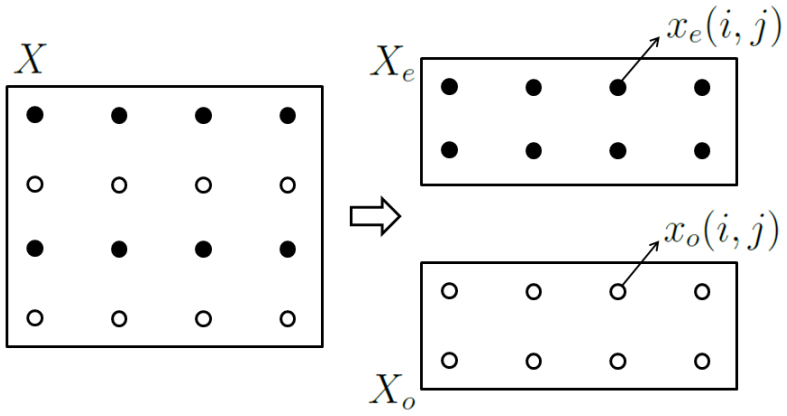


Figure 4.1: Input image and its decomposition.

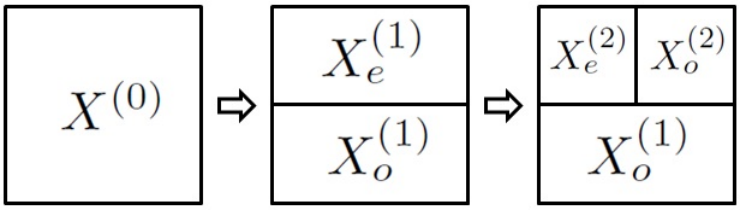
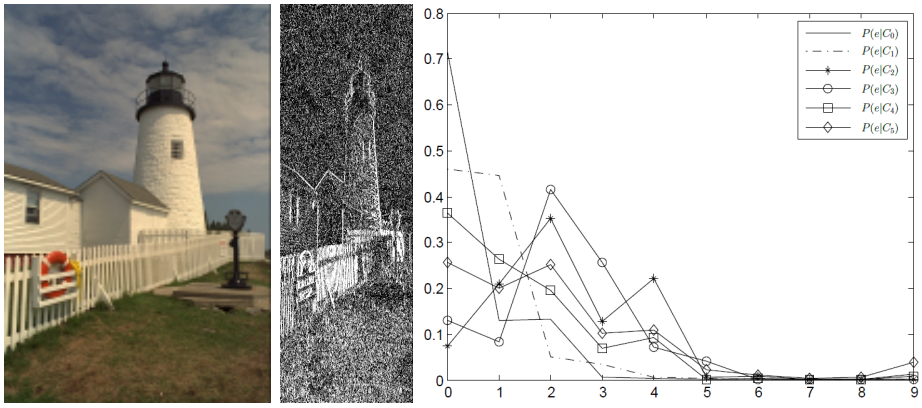


Figure 4.2: Illustration of one level decomposition.



(a) Input image (b) Context (c) Conditional pdf

Figure 4.3: Context modeling (Kodak19).

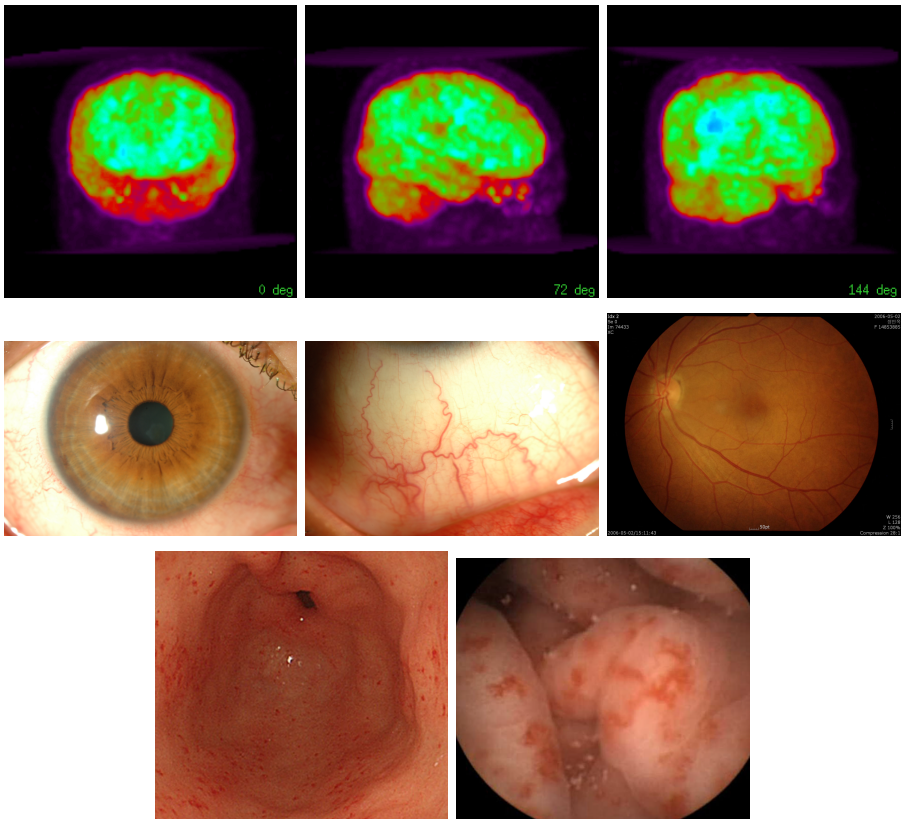


Figure 4.4: The medical images.

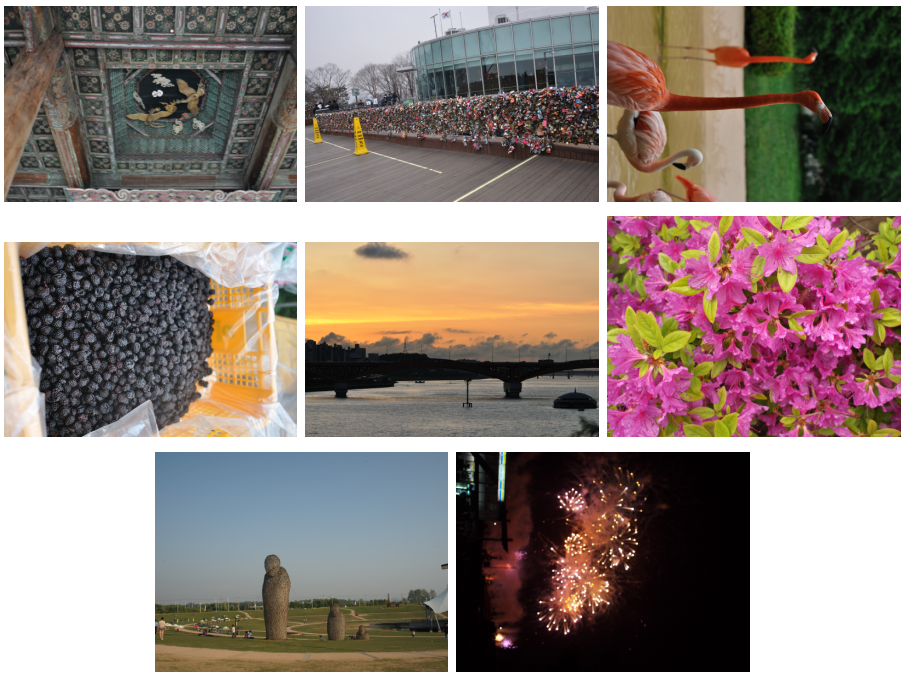


Figure 4.5: The digital camera images.

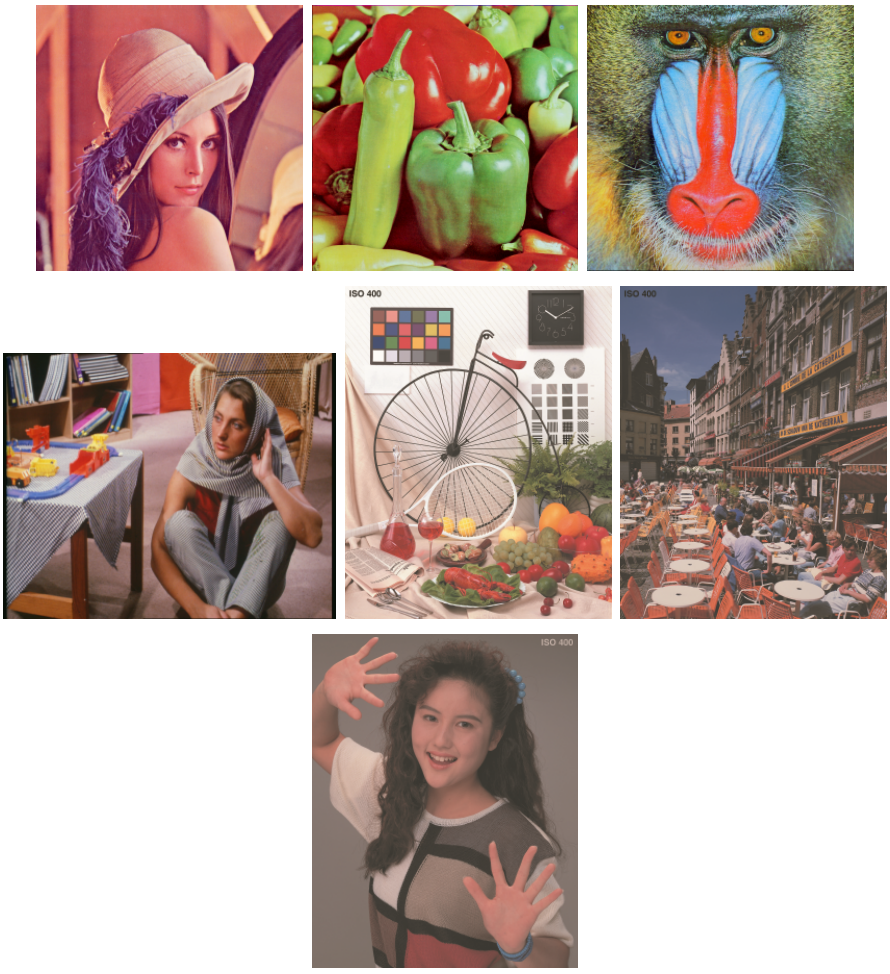


Figure 4.6: The Classic images.

Chapter 5

Color Filter Array Compression

5.1 Introduction

Most digital cameras have a single image sensor plane, where every cell captures wide wavelength range of light. For producing color image with this sensor, green (G), blue (B) or red (R) filters are regularly placed on the cells, which is called color filter array (CFA) sensor. The most widely used pattern of color filter may be the Bayer pattern shown in Fig. 5.1 [12]. Compared with the cameras that capture R, G, and B at each pixel position by separate sensor planes, the CFA cameras need less room for optical system, less power and lower cost. However, since only one of RGB components is available at each cell position, it is required to interpolate other two color components from neighboring data, which is called the demosaic process [13, 14]. Since the advantages of CFA outweigh its disadvantages for compact implementation of color image capturing system, most compact cameras and mobile systems use CFA sensor, and many DSLRs also use CFA.

With the raw data captured by CFA, image processor in the camera performs

most pre-processing steps such as white balancing, denoising, demosaicking, etc, and then the demosaicked RGB image is saved as raw RGB or trasferred to image compressor. Most commercial cameras provide only raw RGB data and/or JPEG-compressed images to the users, and the CFA data are not available. In this case, performing some algorithms on these RGB images is actually doing the processing again on the data which are already processed by image processor in the camera. Hence it deserves consideration to process the untouched CFA data in a powerful computer for better image processing results, instead of doing that in the low-powered image processors of cameras. In terms of compression performances, compressing CFA data instead of demosaicked RGB image is also more efficient as addressed in many literatures on CFA image compression [15, 18–20, 17, 16, 21–23]. Specifically, the compression-first scheme is more efficient than the demosaic-first method because the demosaic process increases the number of data that are somehow correlated. These earlier works include lossy and lossless compression of CFA data, but we will consider only lossless compression in this paper, with the hope that users may consider processing untouched CFA data by complicated algorithms in a powerful PC for producing better images.

The lossy CFA compression methods referenced above [18–20] and some lossless

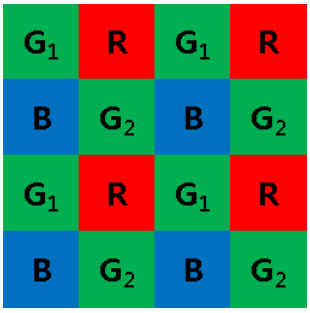


Figure 5.1: Bayer CFA pattern [12].

methods [17, 16] use spectral-spatial transform to generate typical grayscale subimages from a mosaic image, which is followed by standard and/or widely used image coder such as JPEG2000 [7] and JPEG-LS [2]. Specifically, Mallat wavelet packet transform is shown to be beneficial for CFA image compression in [16], and a new spectral-spatial transform is defined for the CFA compression in [17]. However, since the correlation between color components is not sufficiently exploited in the case of transform approaches, more elaborated inter-color prediction methods have also been proposed [21–23]. These methods encode G pixels first and then predict B and R from the encoded G. Especially, the context-matching prediction method [21] provides the best compression rate among the above referenced methods on simulated mosaic data, at the cost of more computational complexity than the transform methods.

In the case of predictive lossless compression methods, accurate prediction is very important for reducing the energy and/or entropy of prediction residual that will be manipulated by the entropy encoder. When there are large prediction errors, which usually arise around the edges or textured areas, the performance of generic arithmetic coder is degraded. However, if we can also estimate the pdf of prediction error for the given situation, compression performance can be further increased by using context-based adaptive encoding scheme as addressed in [4]. More precisely, we can estimate the magnitude of error from the neighboring pixels (context) and thus we can build pdf of error conditioned on the given context. Then even very large errors are well expected ones under some context, and thus do not increase the number of encoded bits. In summary, accurate estimation of error magnitude is as important as accurate pixel prediction in the case of context-adaptive lossless data encoding, and we develop a hierarchical prediction method that can better

predict pixels and also estimate the magnitude of prediction errors for the given neighbors. The information from the neighboring pixels forms the "context," and the conditional pdf of error over the given context is used for efficient entropy coding. In the proposed hierarchical pixel prediction scheme, some of G pixels are used for the prediction of other G, and they are again used for the prediction of R, and then all of these are used for the prediction of B. In this process, already encoded pixels are also used as context for the estimation of conditional pdf for the context-adaptive arithmetic coding. Finally, the prediction errors along with the contexts are encoded by a conventional context adaptive arithmetic coder.

In the experiments, the proposed method is compared with a recent predictive encoding method in [21], which provides the best performance on the simulated CFA data made from KODAK image set [28], and also with the recent transform method in [16]. We also compare the results on some other simulated CFA data and also on real CFA data available at [27]. The comparison shows that the proposed method yields less bpp on all of the images referenced above. The executables of our encoder and decoder are also available at [27].

The rest of this paper is organized as follows. In Section 2, the structure of our encoder is presented. Then Section 3 shows the prediction scheme and Section 4 shows the context modeling for adaptive encoding. Experiments on simulated and real CFA data are presented in Section 5, and the conclusions are given in Section 6.

5.2 Overview of Proposed Encoder

Fig. 5.2 shows the structure of our encoder, which is consisted of hierarchical predictor (inside the dashed block), a conventional grayscale encoder and a context adaptive arithmetic coder. For the input CFA image as in Fig. 5.1, G1 pixels (G pixels in odd rows) are first encoded by a conventional grayscale coder. Second, they are used for the prediction of G2 (G in even rows), which produces the prediction error e_{G2}. Third, G pixels are interpolated to fill in the green values at the positions of R and B pixels. Fourth, these demosaicked G pixels are subtracted from R and B pixels, producing dR and dB. Note that dR and dB are used instead of R and B respectively, for exploiting channel correlation. Fifth, dR values are predicted from the already encoded neighboring dR values, along with the information from G pixels, and the prediction error e_{dR} is obtained. Finally, all the already encoded pixels (including G, dR, and also preceding dB) are used to find an appropriate predictor for a given dB, and the prediction error e_{dB} is generated. The details on the interpolation and prediction methods will be explained in Section 3. The error signals obtained by the prediction block, i.e., e_{G2}, e_{dR}, and e_{dB} are fed to the context-adaptive arithmetic coder.

It needs to be noted that large prediction errors are inevitable near the edges and textured area even if a very elaborated predictor is used, which severely degrades the performance of conventional entropy coder. However, if we can estimate pdf of error for the given neighbors, more efficient encoding is possible as addressed in [4]. More precisely, when we encode an error e_n at a pixel position n, we use the information from the already encoded neighboring pixels as the "context" C_n. In other words, we build the pdf $P(e_n|C_n)$ while encoding, which is used for the adaptive arithmetic

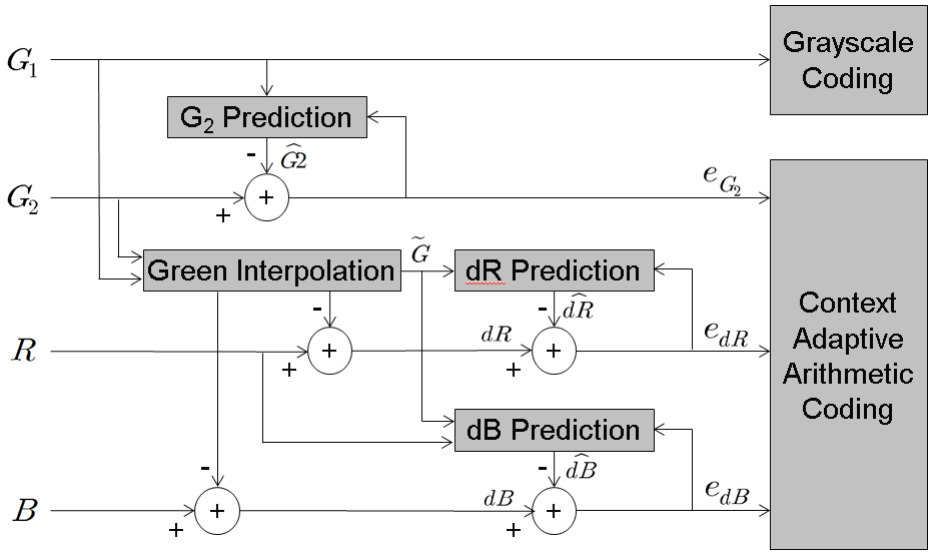
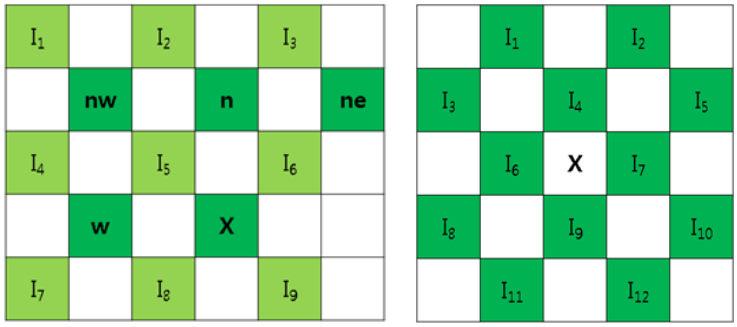


Figure 5.2: Structure of the proposed encoder.

coding. For example, the context in a homogeneous region would be different from the context near the edges, i.e., the pdf for the former case has narrow Laplacian shape and the pdf for the latter will have long tailed shape. Hence, if the context modeling is correct, even the large error is actually a well expected one (with low entropy) and thus the context adaptive encoder produces less bits than a plain entropy coder. The details of context modeling for the CFA data will be explained in Section 4.

5.3 Hierarchical Prediction of CFA data

In this section, we explain the prediction scheme (dashed block in Fig. 5.2) of our encoder. As shown in Fig. 5.1, we define a mosaic image as composed of four subimages: G_1, G_2, R and B. If we denote the mosaic image as X, their relationship can be expressed as



(a) (b)

Figure 5.3: Illustrations for the explanation of (a) prediction of G_2 pixel from the neighboring pixels, and (b) interpolation of missing green pixel (at the position of R and B).

$$\begin{aligned} G_1(i,j) &= X(2i,2j) \\ G_2(i,j) &= X(2i+1,2j+1) \\ R(i,j) &= X(2i,2j+1) \\ B(i,j) &= X(2i+1,2j). \end{aligned} \tag{5.1}$$

As stated previously, we first encode G_1 by a grayscale compression method JPEG-LS, and then hierarchical prediction of other subimages follows.

5.3.1 Prediction of G_2

Let us consider the situation that a pixel denoted as "x" in Fig. 5.3(a) is being encoded. Note that light green boxes are the pixels in G_1 and dark green boxes are the ones in G_2. For the prediction of "x", we can use the already encoded pixels of G_2 such as nw, n, ne, w, and all the neighboring pixels (I_n) in G_1. As in conventional prediction approaches, we define four directional predictors (horizontal, vertical, right diagonal, left diagonal) as

$$\{P_h, P_v, P_{df}, P_{dl}\} = \left\{w, n, \frac{I_5 + I_9}{2}, \frac{I_6 + I_8}{2}\right\}, \tag{5.2}$$

which are consisted of the nearest pixel(s) to the "x", into the corresponding directions. In the proposed prediction scheme, we do not choose one out of these four predictors, but we choose two best predictors and combine them with appropriate weights as will be discussed later.

The choice of predictors and weights is of course based on the direction of edges around the "x". We define the edge directivity around "x" as

$$D_a = \frac{\sum(\text{pixel differences around x into the direction } a)}{\sum(\text{distances of pixel pairs})} \tag{5.3}$$

where $a = h$, v, dr or dl. By this definition, small D_a means that edge is probably into the direction of a and thus predictor P_a would be a good choice. To be specific, four edge directivities around the "x" in Fig. 5.3(a) are defined as

$$\begin{aligned}
D_h &= (|nw - n| + |I_4 - I_5| + |I_5 - I_6| \\
&\quad + |I_7 - I_8| + |I_8 - I_9|)/10 \\
D_v &= (|nw - w| + |I_2 - I_5| + |I_5 - I_8| \\
&\quad + |I_3 - I_6| + |I_6 - I_9|)/10 \\
D_{dr} &= [(|nw - I_5| + |n - I_6| + |w - I_8| \\
&\quad + |I_5 - I_9|)/5\sqrt{2}] \times \alpha \\
D_{dl} &= [(|n - I_5| + |ne - I_6| + |I_5 - w| \\
&\quad + |I_6 - I_8|)/5\sqrt{2}] \times \alpha.
\end{aligned} \tag{5.4}$$

Note that the number of considered pixel pairs for D_h and D_v is 5, but the sum of

distances is 10 because each pair is two pixels away. We may consider less number of pixel pairs around "x," but we choose to include all the adjacencies for reducing noisy results. Conversely, considering too many pixel pairs beyond this bound may not reflect the correct edge direction. In the case of D_{dl} and D_{dr}, the number of considered pixel pairs is 4. But since one of the pairs is two pixels away into the diagonal direction ($|I_5 - I_9|$ or $|I_6 - I_8|$ which are paired across "x"), the total sum of distances is to be $5\sqrt{2}$. Also note that we intentionally reduce D_{dl} and D_{dr} by multiplying α (< 1) to them as shown in the above equation, with the intention that diagonal predictors are more often selected than horizontal or vertical predictors. The reason for this is from the observation of Fig. 5.3(a) and eq. (5.2), which shows that P_{dr} and P_{dl} are consisted of $\{I_5, I_9\}$ or $\{I_6, I_8\}$ which are adjacent to "x," whereas $P_h = w$ and $P_v = n$ are two pixels away from "x," and thus the diagonal predictors are more often close to "x" than the horizontal or vertical predictors. In all the experiments that will be discussed in Section 5, the α is set to 0.2.

As stated above, the actual prediction of "x" is obtained as a weighted sum of two predictors among eq.(5.2). Denoting the smallest and the second smallest D_a's in eq. (5.4) as D_1 and D_2, the weights for the prediction of "x" are defined as

$$w_1 = D_1 + 1 \quad \text{and} \quad w_2 = D_2 + 1, \tag{5.5}$$

where 1 is added to avoid "divide by zero". Then the actual estimator of "x" is defined as

$$\hat{x} = \frac{w_2 P_1 + w_1 P_2}{w_1 + w_2}, \tag{5.6}$$

where P_1 and P_2 are the predictors corresponding to the directions of D_1 and D_2

respectively.

5.3.2 Interpolation of green values in the positions of R and B

As stated above, the blue and red pixels are not directly encoded, but $dR = R - G$ and $dB = B - G$ are predicted and encoded. For this, we need to find the green values in the positions of R and B. This can be done just by using any of conventional demosaic algorithms, but we use the same method as G2 is predicted from G1. Specifically, let us consider the situation in Fig. 5.3(b), where the missing green pixel "x" is interpolated using the neighboring pixels. Since all the closest 4 neighbors are available in this case, we interpolate (predict) only into horizontal and vertical directions as

$$\{P_h, P_v\} = \left\{ \frac{I_6 + I_7}{2}, \frac{I_4 + I_9}{2} \right\}. \tag{5.7}$$

We also calculate the edge directivity in the same manner of eq. (5.4) as

$$\begin{aligned} D_h &= |I_1 - I_2| + |I_3 - I_4| + |I_4 - I_5| + |I_6 - I_7| \\ &\quad + |I_8 - I_9| + |I_9 - I_{10}| + |I_{11} - I_{12}| \\ D_v &= |I_3 - I_8| + |I_1 - I_6| + |I_6 - I_{11}| + |I_4 - I_9| \\ &\quad + |I_2 - I_7| + |I_7 - I_{12}| + |I_5 - I_{10}| \end{aligned}, \tag{5.8}$$

where the divisors are omitted because they are the same. Then the final interpolator is defined as

$$\hat{x} = \frac{w_v P_h + w_h P_v}{w_h + w_v} \tag{5.9}$$

where

$$w_h = D_h + 1 \quad \text{and} \quad w_v = D_v + 1. \tag{5.10}$$

5.3.3 Prediction of red and blue pixels

After obtaining green values in the positions of R and B, we can compute dR and dB. First, let us consider the encoding of dR as illustrated in Fig. 5.4(a). For predicting the dR in the position marked "x", the predictors are simply the neighboring dR's in the corresponding directions as

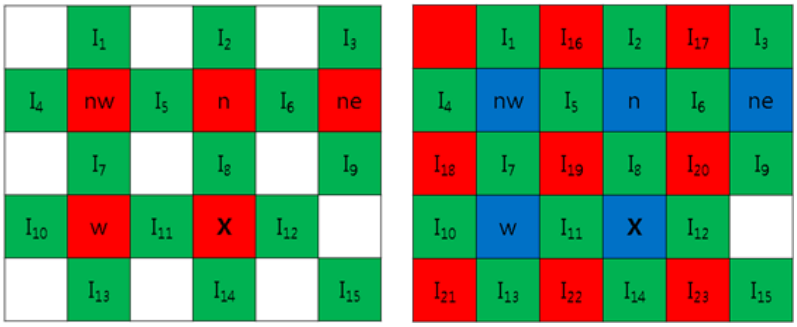
$$\{P_h, P_v, P_{dr}, P_{dl}\} = \{w, n, nw, ne\}, \tag{5.11}$$

In a similar manner as the predictors of G2 are selected, we define the edge directivity for dR by using all the neighboring G and R pixels as

$$\begin{aligned}
D_h &= (|nw - n| + 4\,|I_{10} - I_{11}| + 4\,|I_{11} - I_{12}| \\
&\quad +2\,|I_7 - I_8| + 2\,|I_{13} - I_{14}|)/26 \\
D_v &= (|nw - w| + 4\,|I_2 - I_8| + 4\,|I_8 - I_{14}| \\
&\quad +2\,|I_5 - I_{11}| + 2\,|I_6 - I_{12}|)/26 \\
D_{dr} &= (2\,|I_5 - I_8| + 2\,|I_7 - I_{11}| + |I_8 - I_{12}| \\
&\quad + |I_{11} - I_{14}|)/6\sqrt{2} \\
D_{dl} &= (|n - w| + 2\,|I_6 - I_8| + 2\,|I_9 - I_{12}| \\
&\quad + |I_8 - I_{11}| + |I_{12} - I_{14}|)/8\sqrt{2}
\end{aligned} \tag{5.12}$$

where closer pixels to "x" are given larger weights. The final predictor is obtained in the same manner of eqs. (5.5) and (5.6) as

Encoding of dB is performed in the same way, except that red pixels are also



(a) (b)

Figure 5.4: (a) Prediction of dR pixels, and (b) prediction of dB pixels.

involved in defining the edge directivity. Consider the situation that a dB in "x" of Fig. 5.4(b) is being encoded. The directional predictors are defined the same as eq. (5.11), and directivities can be more efficiently computed by using the previously calculated ones as

$$\begin{aligned} D_h &= (26D_h^R + 2\,|I_{18} - I_{19}| + 2\,|I_{19} - I_{20}| \\ &\quad +2\,|I_{21} - I_{22}| + 2\,|I_{22} - I_{23}|)/42 \\ D_v &= (26D_v^R + 2\,|I_{16} - I_{19}| + 2\,|I_{19} - I_{22}| \\ &\quad +2\,|I_{17} - I_{20}| + 2\,|I_{20} - I_{23}|)/42 \\ D_{dr} &= (6\sqrt{2}D_{dr}^R + |I_{19} - I_{23}|)/8\sqrt{2} \\ D_{dl} &= (8\sqrt{2}D_{dl}^R + |I_{20} - I_{22}|)/10\sqrt{2} \end{aligned} \tag{5.13}$$

where D_a^R ($a \in \{h, v, dr, dl\}$) refers to D_a for the red pixels in (5.12). The final prediction of dB is also calculated as eqs. (5.5) and (5.6).

5.4 Encoding Prediction Errors

This section presents the "context adaptive arithmetic coding" block of our encoder in Fig. 5.2. In the case of predictive encoders, it is important to reduce the energy of residual signal as much as possible. In addition, if we correctly estimate pdf of residual signal, we can further reduce entropy and thus enhance the compression performance by using context-based coding as addressed in [4]. Hence, we estimate the pdf of prediction errors, from the states of neighboring pixels (context) in the form of conditional pdf.

In the case of image encoder, it is well expected that the prediction errors have large magnitude near the edges or textured area, and they are also spatially correlated. Based on this, we introduce three basic assumptions for the context modeling. First, the prediction errors are spatially correlated, i.e., a pixel neighboring the ones with large residuals has the high possibility of having large prediction error. Second, error magnitude is proportional to the gradients in the neighborhood. Last, the predictors into four different directions will have similar values when a pixel is located in smooth region. Based on these assumptions, the magnitude of prediction error is estimated by exploiting the related parameters that have already been computed in the process of prediction. To be precies, the magnitude is estimated as

$$\delta = E\left[|e|\right] + 2D_1 + STD(P), \tag{5.14}$$

where $E\left[|e|\right]$ is the expectation of absolute prediction errors in the neighborhood ($M \times M$ block of pixels left and upper to the x), D_1 is the smallest edge direction measure among $\{D_h, D_v, D_{dr}, D_{dl}\}$ for the corresponding color, and $STD(P)$ means standard deviation of the four directional predictors $\{P_h, P_v, P_{dr}, P_{dl}\}$. For testing

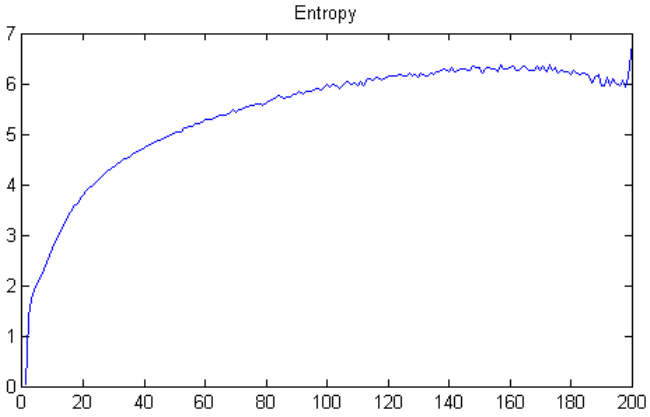


Figure 5.5: Entropy of resiual with respect to δ.

the validity of this estimator, we plot the entropy of prediction errors with respect to δ in Fig. 5.5 using the images from Kodak set [28]. It can be seen that the entropy increases almost monotonically as δ increases, and thus δ is a good measure of error magnitude.

The estimator of prediction residual δ is quantized into K steps as

$$C_n : q_{n-1} \leq \delta < q_n, \text{ for } n = 1, \cdots, K \tag{5.15}$$

with $q_0 = 0$ and $q_K = \infty$, and each step C_n represents the "context" or state of the pixel to be encoded. In all the experiments, K is set to 30 and the quantizer is set as shown in Table xxx, which is designed such that each bin has almost the same number of samples when tested with the Kodak image set. With this context model, a generic adaptive arithmetic coder [24] is used to encode the prediction errors, which learns $P(e|C_n)$ while encoding.

5.5 Experimental Results

5.5.1 The Proposed Method

The proposed method is compared with several standard and/or widely used lossless image compression methods such as JPEG2000 [7] and JPEG-LS [2], and also with recently introduced CFA lossless compression methods: LCMI [16], and CMBP [21]. In all the experiments, we use JPEG-LS for the compression of G_1, and the parameters are set as $\alpha = 0.2$, $K = 30$, and the quantizer for δ is set as Table xxx. Actually, the compression performance is not much affected by the quantization parameter setting, because the arithmetic coder works adaptively to image statistics. The parameter α is decided from the test of Kodak image set, and used the same for all the other images in the experiments.

First, we show the results on 1024×768 real CFA images, captured with Sony ICX204 Progressive scan CCD. All the images used in this paper, along with the executables of encoder and decoder, are avaiable in [27]. Some of cropped parts of the images are also shown in Fig. xxx. The comparison results are shown in Table 5.1, where it can be seen that the proposed method gives the least bpp. To be specific, the proposed method further reduces the number of bits by 20.77%, 4.97%, 3.59%, and 9.09% over JPEG-LS, JPEG2000, LCMI, and CMBP respectively.

Second, simulated Kodak CFA images are tested, which are widely used for the test of demosaic algorithms and also for the test of lossy/lossless compression algorithms. The "simulated" CFA image means that it is sampled from the original RGB image in the Bayer pattern. The results are shown in Table 5.2, which shows that the proposed method reduces the bits by 23.96%, 9.80%, 7.49%, and 1.85% over JPEG-LS, JPEG2000, LCMI, and CMBP respectively.

Table 5.1: Comparison of the algorithms for a set of real CFA images.

	JPEG-LS	JPEG2K	LCMI	CMBP	Proposed
1	6.4208	5.3910	5.2551	4.9955	**4.9097**
2	6.4222	5.4852	5.3288	5.0880	**4.9971**
3	6.5599	5.3856	5.2279	4.9952	**4.9080**
4	5.5381	4.5526	4.4922	4.2554	**4.1369**
5	5.5221	4.4454	4.3906	4.1666	**4.0470**
6	5.2284	4.6128	4.5490	4.3905	**4.1036**
7	4.8082	3.8070	3.7007	3.6133	**3.3653**
8	4.7686	3.7243	3.6204	3.5445	**3.2835**
9	5.5002	4.7357	4.7090	4.4703	**4.2842**
10	5.5565	4.7316	4.7025	4.4553	**4.2689**
11	4.8489	3.9749	3.9521	3.7361	**3.5367**
12	4.7804	3.9242	3.9090	3.7019	**3.4964**
13	4.6067	3.8626	3.8749	3.6805	**3.4618**
14	4.8703	4.1102	4.1070	3.9138	**3.6962**
Avg.	5.3880	4.4817	4.4157	4.2148	4.0354

Lastly, we also test the algorithms on high-resolution simulated CFA images from commercial digital cameras such as NIKON D90 (4288×2848) and OLYMPUS E-P1 (4032×3024), and the results are summarized in Table 5.3 and Table 5.4. Note that we can extract original (real) CFA data from the camera module that we mentioned above (Sony ICX204), but the commercial digital cameras usually do not provide the CFA image so that only simulated data could be tested. For NIKON D90 images, the proposed method shows coding gain by 34.34%, 14.77%, 7.63%, and 4.47% over JPEG-LS, JPEG2000, LCMI, and CMBP respectively. For OLYMPUS E-P1 images, the proposed method outperforms by 41.60%, 17.99%, 6.74%, and 4.57% over JPEG-LS, JPEG2000, LCMI, and CMBP respectively.

Finally, it is interesting to see that JPEG2000 shows much better performance than JPEG-LS when applied to CFA data, sometimes comparable to LCMI [16]. (Actually, it is not fair to compare JPEG2000 and JPEG-LS with the CFA-specific

Table 5.2: Comparison of the algorithms for a set of simulated CFA images from KODAK set.

	JPEG-LS	JPEG2K	LCMI	CMBP	Proposed
1	6.4726	5.8105	5.7592	5.4701	**5.4696**
2	7.0514	5.1403	4.6862	4.3341	**4.2862**
3	5.8519	4.2090	4.0223	3.7360	**3.6990**
4	6.5800	4.9483	4.6839	4.3965	**4.3530**
5	6.5132	5.9617	5.8844	5.4024	**5.3648**
6	6.1042	5.2556	5.0810	4.8398	**4.8308**
7	5.8027	4.5055	4.3218	3.9339	**3.8636**
8	6.2231	5.9045	5.8981	5.5941	**5.5775**
9	5.0830	4.4044	4.3519	4.1964	**4.1381**
10	5.4077	4.5614	4.4803	4.2359	**4.1911**
11	5.8526	5.0236	4.9666	4.6669	**4.6602**
12	5.6290	4.5156	4.3306	4.0772	**4.0447**
13	6.8322	6.4079	6.3991	6.1398	**6.1162**
14	6.4785	5.5865	5.4927	5.1553	**5.1278**
15	5.8260	4.6027	4.4627	4.1299	**3.9941**
16	5.6168	4.5929	4.5142	**4.3404**	4.3473
17	4.9583	4.5509	4.5538	4.2812	**4.2282**
18	6.1751	5.5864	5.5679	5.2921	**5.2496**
19	5.4830	4.9102	4.8716	4.7102	**4.6668**
20	4.4684	4.1185	3.9868	3.5773	**3.2277**
21	5.9818	5.1132	4.9539	4.7784	**4.7279**
22	6.4901	5.2505	5.0924	4.8464	**4.8075**
23	6.5814	4.5033	4.0401	3.8760	**3.7614**
24	5.7003	5.2270	5.2845	4.9126	**4.7910**
Avg.	5.9651	5.0288	4.9036	4.6218	4.5635

algorithms. The comparison is just for reference how much gain would be obtained by the CFA-specific methods over the conventional compressors.) It is because the discrete wavelet transform in JPEG2000 plays a role of spectral-spatial transform, like the transform in LCMI.

Table 5.3: Comparison of the algorithms for a set of simulated CFA images from digital cameras.

	JPEG-LS	JPEG2K	LCMI	CMBP	Proposed
1	4.9348	3.9525	3.6783	3.6629	**3.5470**
2	6.2309	5.2131	4.9453	4.6696	**4.5349**
3	4.8743	3.8237	3.6897	3.5212	**3.4322**
4	5.2951	3.5034	3.1116	3.0861	**2.9388**
5	5.0022	3.8424	3.4889	3.3515	**3.1916**
6	4.4009	3.3467	2.9369	2.8354	**2.7094**
Avg.	5.1230	3.9470	3.6418	3.5211	3.3923

5.5.2 Demosaic-first and Compression-first Schemes

Comparison of demosaic-first and compression-first Schemes is performed both for lossless and lossy compression. At first, lossless bitrates of the two schemes are compared. 24 Kodak images are sampled with Bayer CFA pattern to simulate CFA image. The simulated CFA images are compressed by the proposed CFA compression method which will be presented in Chapter 5. As demosaic-first scheme, the simulated CFA images are first demosaicked by Alternating Projection (AP) method [14], and the demosaicked images are compressed by standard color images compression method JPEG2000 (Lossless mode). The lossless bitrates are presented in Table5.5. The compression-first scheme considerably outperforms the conventional demosaic-first method. This is mainly because the number of samples for demosaic-first scheme is increase three times through demosaicking process but it is hard for lossless compression to effectively remove the increased redundancy.

Rate-distortion performances of lossy versions of both schemes are also examined. Test for demosaic-first scheme is the same as the lossless test described above except that JPEG2000 runs in lossy mode. To be specific, demosaicked images of the simulated CFA images from Kodak set are compressed by JPEG2000 (Lossy mode).

Table 5.4: Lossless bit rates of various schemes (bpp) for the simulated CFA images from OLYMPUS E-P1 set (4032×3024 resolution).

	JPEG-LS	JPEG2K	LCMI	CMBP	Proposed
1	7.2529	5.2788	4.6280	4.3925	**4.2708**
2	5.9338	4.6118	4.1577	4.0615	**3.9066**
3	7.2368	5.2639	4.5766	4.5273	**4.3621**
4	7.2347	5.0365	4.2904	4.2363	**4.0692**
5	7.1544	4.5750	3.4266	3.3791	**3.1702**
6	6.2255	4.9155	4.5286	4.3862	**4.2805**
7	7.3168	4.4518	3.2780	3.3215	**3.1100**
8	7.4666	5.7251	5.2342	5.0616	**4.9478**
9	6.8089	4.7928	3.9954	3.9435	**3.7522**
10	5.9278	4.5701	4.2821	4.1925	**4.0764**
11	5.4718	3.9601	3.5948	3.5035	**3.3704**
12	4.6019	3.2978	3.0417	3.0045	**2.8533**
13	6.8542	4.9610	4.3682	4.3751	**4.2804**
14	6.0079	4.5197	4.2131	4.1847	**4.0922**
15	6.1879	4.5253	4.1442	4.1652	**4.0618**
16	6.9412	4.9115	4.2843	4.2995	**4.1944**
17	6.3583	4.1038	3.2204	3.0957	**2.9096**
18	6.8115	4.3326	3.4836	3.2540	**3.0803**
19	4.7375	3.2598	2.9812	2.9254	**2.7778**
20	4.9997	3.4218	3.1413	3.0289	**2.8669**
21	4.7266	3.0694	2.7927	2.7388	**2.5508**
22	4.5890	3.3026	3.0746	2.9699	**2.8089**
23	4.5414	3.9133	3.8405	3.6271	**3.4869**
24	5.4500	3.7675	3.3707	3.1868	**3.0213**
Avg.	6.1182	4.3570	3.8312	3.7442	3.5959

To measure the quality of decompressed image, color peak signal to noise ratio (CPSNR) is calculated regarding the demosaicked images as original. CPSNR is generally used quality measure for color image and defined as

$$\mathrm{CPSNR} = 10 \log_{10} \frac{255^2}{\frac{1}{3N_1N_2} \sum_X \sum_{n_1} \sum_{n_2} \left(\hat{X}(n_1, n_2) - X(n_1, n_2) \right)^2}, \tag{5.16}$$

Table 5.5: Lossless bit rates of demosaic-first and compression-first schemes (bpp) for the simulated CFA images (Kodak).

	Demosaic-first	Compression-first
1	10.1656	5.4342
2	8.1022	4.2609
3	7.1468	3.6769
4	8.0404	4.3515
5	10.3446	5.3299
6	8.9909	4.7855
7	7.6102	3.8357
8	10.8945	5.5059
9	7.6824	4.1130
10	7.8200	4.1531
11	8.6956	4.6190
12	7.6509	4.0044
13	11.4540	6.0769
14	9.5519	5.0984
15	7.9850	3.9714
16	7.9589	4.3067
17	7.8759	4.2177
18	9.7637	5.2398
19	8.6755	4.6460
20	6.9223	3.1886
21	8.7850	4.6962
22	8.9710	4.7949
23	7.2234	3.7582
24	9.3846	4.7588
Avg.	8.6540	4.5343

where N_1 and N_2 denote image dimensions, $n_1 = 1, 2, ..., N_1$ and $n_2 = 1, 2, ..., N_2$ denotes pixel coordinates, and $X \in \{R, G, B\}$ denotes the color channels.

For the test of lossy version of compression-first scheme, LCMI method is modified and used. Input CFA image is decomposed by discrete wavelet transform (DWT) in lossless manner and the four subband images are independently encoded with lossy JPEG2000 which is different part from the original LCMI. To measure CPSNR, the compressed code is decoded into CFA image, and the decompressed im-

Table 5.6: Lossy bit rates and CPSNRs of demosaic-first and compression-first schemes for the simulated CFA images (Kodak).

Demosaic-first		Compression-first	
BPP	CPSNR	BPP	CPSNR
1.4894	38.7030	1.4557	36.8358
1.9891	40.9446	1.9537	39.5055
2.4885	42.9042	2.4545	41.8539
2.9888	44.6837	2.9530	43.8910
3.4888	46.3000	3.4527	45.8239
3.9890	47.8276	3.9540	47.7306
4.4547	49.0867	4.4354	49.9097
5.0328	50.5131	5.1024	53.9637

age is demosaicked by AP method. The demosaicked image from the decompressed CFA image is used as $\hat{X}$ in (5.16), and the demosaicked image from the simulated CFA image is used as X in the same equation.

Rate-distortion (RD) curves of the demosaic-first and the compression-first schemes are shown in Fig. 5.10. In lossy compression, bitrate and image quality are in trade-off relation, so RD curve is useful to compare performances of different methods. When a curve is located above another one, it is said to be better in RD performance because it shows better performance in quality measure on the same bitrate. From Fig. 5.10, demosaic-first scheme is better in the lower bitrate than about 4 bpp, and compression-first scheme is better in the rest region. Considering that the lossless bitrate for the same input was 4.5343 bpp, however, the region where lossy version of compression-first scheme is useful is between 4 bpp and 4.5 bpp approximately.

Demosaic-first scheme outperforms compression-first scheme in practically useful bpp region for lossy compression. It is first because lossy compression effectively removes redundancy which is introduced by demosaicking. And because demosaicking is non-linear process, compression artifacts to CFA image might cause large demo-

saicking error in compression-first scheme. Therefore, CFA image is more suitable for lossless compression.

5.6 Conclusion

We have proposed a new lossless compression algorithm for the Bayer-patterned CFA images. The proposed scheme is to predict the color components hierarchically and to use context adaptive arithmetic coding. The hierarchical prediction means that we first encode half of green samples by a conventional grayscale coder, and use these samples for the prediction of other half green samples. Then all the green samples are used for the prediction of red ones, all of which are then used for the blue prediction. In deciding the predictors, edge directivity is considered for reducing the prediction residual. The information obtained in the prediction process is also used for the context modeling of prediction residual. The proposed algorithm is tested on the simulated data that have been popularly used in the literatures, and also on real CFA and some additional high-resolution simulated CFA data. The results show that the proposed method yields less bpp than the transform-based method and other existing prediction-based method.

Figure 5.6: The Kodak images.

Figure 5.7: The real CFA images (demosaicked).

Figure 5.8: The NIKON D90 images.

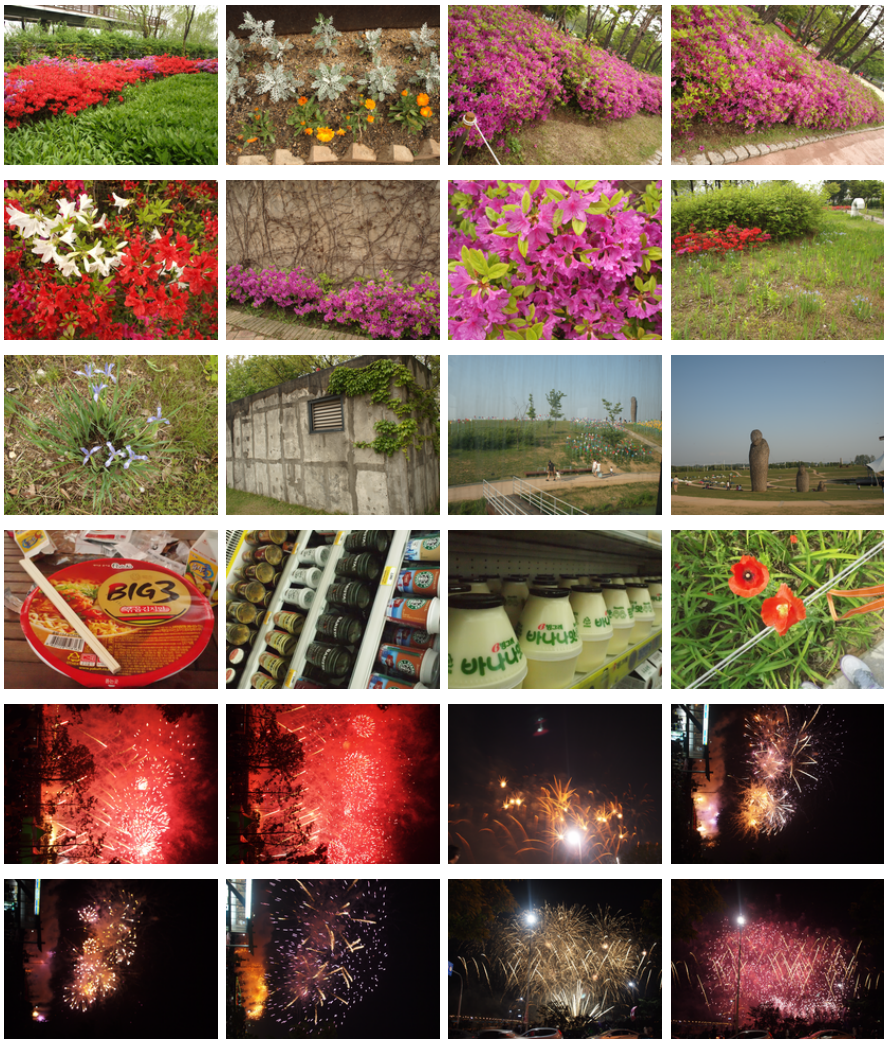


Figure 5.9: The OLYMPUS E-P1 images.

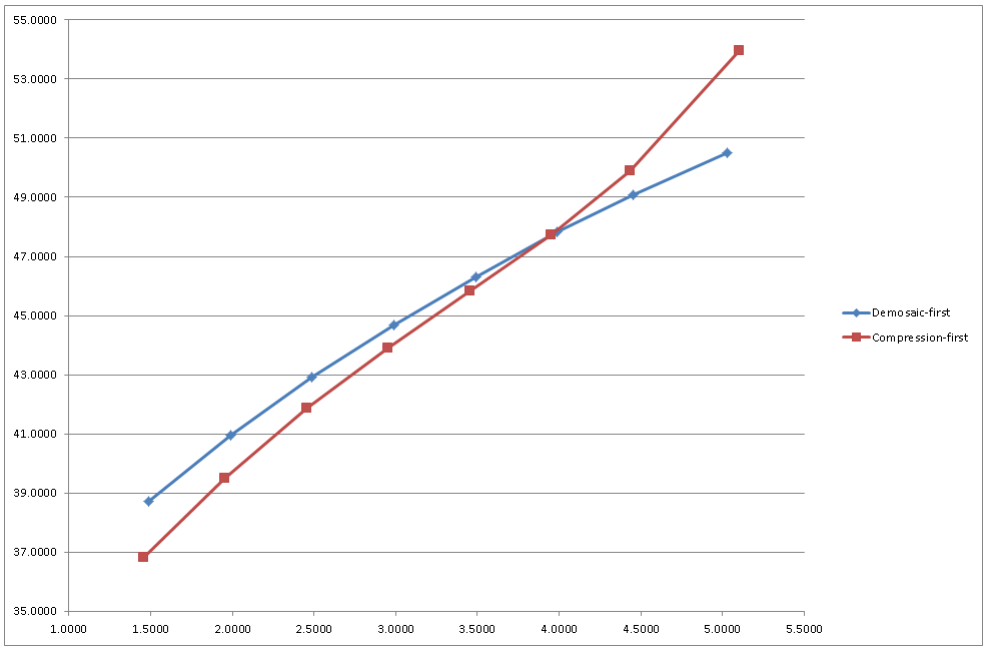


Figure 5.10: Rate-distortion performances of lossy color image compression (demosaic-first) and lossy CFA compression (compression-first). The x-axis denotes bit per pixel (bpp) and the y-axis denotes CPSNR.

Chapter 6

Conclusions

Lossless image compression is less used than lossy compression due to its large memory or bandwidth requirements. However, lossless image compression is indispensable for some fields, such as medical, prepress, scientific, and artistic areas, and people desiring the technique are increasing due to advances in cheaper and bigger storage devices. In this dissertation, algorithms using interchannel correlation are studied, with which lossless compression schemes for color image and color filter array image are proposed.

In Chapter 3, a new reversible color transform is proposed, which consists of the conventional RCT and additional lifting steps to decorrelate chroma images Cu and Cv further. Nearly optimal but simple parameters are found, and the proposed scheme shows comparable decorrelation performance with YC_bC_r which can be applied only to lossy compression. In addition, lossless bit rates of standard lossless images coder are presented, in which the proposed RCT outperforms the conventional RCT over 1.26% with minimum increase in operation.

In Chapter 4, a lossless color image compression method based on a new hier-

archical encoding scheme is proposed. Specifically, an input RGB image is transformed into $YC_u'C_v'$ color space using the new RCT. After the color transformation, the luminance channel Y is compressed by a conventional lossless image coder. The chrominance channels are encoded with the proposed hierarchical decomposition and directional prediction. Finally, an appropriate context modeling of prediction residuals is introduced and generic arithmetic coding is applied. The proposed method and several conventional methods are tested on the Kodak image set, some medical images, and digital camera images, and it is shown that average file size reductions over JPEG2000 for these sets are 5.85%, 10.40%, and 4.89% respectively. When the mode selection is tried, further encoding gain can be obtained.

At last, a new lossless compression algorithm for the Bayer-patterned CFA images is proposed in Chapter 5. The proposed scheme is to predict the color components hierarchically and to use context adaptive arithmetic coding. The hierarchical prediction means that we first encode half of green samples by a conventional grayscale coder, and use these samples for the prediction of other half green samples. Then all the green samples are used for the prediction of red ones, all of which are then used for the blue prediction. In deciding the predictors, edge directivity is considered for reducing the prediction residual. The information obtained in the prediction process is also used for the context modeling of prediction residual. The proposed algorithm is tested on the simulated data that have been popularly used in the literatures, and also on real CFA and some additional high-resolution simulated CFA data. The results show that the proposed method yields less bpp than the transform-based method and other existing prediction-based method.

The proposed three methods, new RCT, hierarchical prediction scheme, and efficient context modeling, are applied to lossless color image compression and lossless

CFA image compression. Each of them was test in various datasets and outperformed all the benchmark methods.

Bibliography

[1] W. B. Pennebaker and J. L. Mitchell, "JPEG Still Image Data Compression Standard," New York: Van Nostrand Reinhold, 1993.

[2] Information Technology–Lossless and Near-Lossless Compression of Continuous-Tone Still Images (JPEG-LS), ISO/IEC Standard 14495-1, 1999.

[3] M. Weinberger, G. Seroussi, and G. Sapiro, "The LOCO-I lossless image compression algorithm: principles and standardization into JPEG-LS," IEEE Trans. Image Process., vol. 9, no. 8, pp. 1309-1324, Aug. 2000.

[4] X. Wu and N. Memon, "Context-based, adaptive, lossless image coding," IEEE Trans. Commun., vol. 45, no. 4, pp.437 - 444, Apr. 1997.

[5] X. Wu and N. Memon, "Context-based lossless interband compression–Extending CALIC," IEEE Trans. Image Process., vol. 9, pp. 994-1001, Jun. 2000.

[6] E. Magli, G. Olmo, E. Quacchio, "Optimized on-board lossless and near-lossless compression of hyperspectral data using CALIC," IEEE Geoscience and Remote Sensing Letters, Vol. 1, No. 1, pp. 21-25, Jan. 2004.

[7] Information Technology–JPEG 2000 Image Coding System–Part 1: Core Coding System, INCITS/ISO/IEC Standard 15444-1, 2000.

[8] C. Poynton, "A Guided Tour of Color Space," SMPTE Advanced Television and Electronic Imaging Conference, San Fransisco, pp. 167-180, 1995.

[9] Digital Compression and Coding of Continuoustone Still Images, Part 1, Requirements and Guidelines. ISO/IEC JTC1 Draft International Standard 10918-1, Nov. 1991.

[10] ISO/IEC JTC 1, "Coding of audio-visual objects – Part 2: Visual," ISO/IEC 14496-2 (MPEG-4 Part 2), Jan. 1999.

[11] ITU-T and ISO/IEC JTC 1, "Advanced Video Coding for generic audio-visual services," ITU T Rec. H.264 and ISO/IEC 14496-10 (AVC), May 2003 (and subsequent editions).

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

[13] B. K. Gunturk, J. W. Glotzbach, Y. Altunbasak, R. W. Schafer, and R. M. Mersereau, "Demosaicking: Color filter array interpolation," IEEE Signal Process. Mag., vol. 22, no. 1, pp. 44-54, Jan. 2005.

[14] B. K. Gunturk, Y. Altunbasak, and R. M. Mersereau, "Color plane interpolation using alternating projections," IEEE Trans. Image Process., vol. 11, no. 9, pp. 997-1013, Sep. 2002.

[15] N. X. Lian, L. Chang, V. Zagorodnov, and Y. P. Tan, "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.

[16] 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.

[17] H. S. Malvar, and G. J. Sullivan, "Progressive-to-lossless compression of color-filter-array images using macropixel spectral-spatial transformation," Data Compression Conference (DCC), 2012.

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

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

[20] R. Lukac and K. N. Plataniotis, "Single-sensor camera image compression," IEEE Trans. Consum. Electron., vol. 52, no. 2, pp. 299-307, 2006.

[21] 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.

[22] K. H. Chung and Y. H. Chan, "A Fast Reversible Compression Algorithm for Bayer Color Filter Array Images," Proceedings on Asia-Pacific Signal and Information Processing Association (APSIPA), 2009.

[23] 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. 637649, Jul. 2012.

[24] A. Said, "Arithmetic Coding," in Lossless Compression Handbook, K. Sayood, ed., Academic Press, San Diego, CA, 2003.

[25] N. X. Lian, L. Chang, V. Zagorodnov, and Y. P. Tan, "Reversing Demosaicking and Compression in Color Filter Array Image Processing: Performance Analysis and Modeling," Trans. Image Process., vol. 15, no. 11, pp. 3261-3278, Nov. 2006.

[26] N. Zhang and X. Wu, "Lossless Compression of Color Mosaic Images," IEEE Trans. Image Process., vol. 15, no. 6, pp. 1379-1388, Jun. 2006.

[27] http://ispl.snu.ac.kr/light4u/project/CFAcompression

[28] http://www.site.uottawa.ca/ edubois/demosaicking, images from KODAK Photo CD Photo Sampler, 1991.

[29] Z. Mai, P. Nasiopoulos, and R. Ward, "A wavelet-based intra-prediction lossless image compression scheme," International Conference on Consumer Electronics, Jan. 2009.

[30] H. S. Malvar and G. J. Sullivan, "Progressive-to-Lossless Compression of Color-Filter-Array Images Using Macropixel Spectral-Spatial Transformation," Data Compression Conference (DCC), Apr. 2012.

[31] N. Zhang and X. Wu, "Lossless compression of color mosaic images," IEEE Trans. Image Process., Vol. 15 , no. 6, pp. 1379-1388, Jun. 2006.

[32] B. K. Gunturk, Y. Altunbasak, and R. M. Mersereau, "Color plane interpolation using alternating projections," IEEE Trans. Image Process., vol. 11, no. 9, pp. 997-1013, Sep. 2002.

[33] B. Bross, W.-J. Han, G. J. Sullivan, J.-R. Ohm, and T. Wiegand, "High efficiency video coding (HEVC) text specification draft 8," ITU-T/ISO/IEC Joint Collaborative Team on Video Coding (JCT-VC) document JCTVC-J1003, Jul. 2012.

[34] H. S. Malvar, G. J. Sullivan, and S. Srinivasan, "Lifting-based reversible color transformations for image compression," Proc. SPIE Applications of Digital Image Processing XXXI, San Diego, CA, USA, Vol. 7073, p. 707307.1-707307.10, 2008.

[35] G. Sullivan, "Approximate Theoretical Analysis of RGB to YCbCr to RGB Conversion Error," ISO/IEC JTC1/SC29/WG11 and ITU-T SG16 Q.6 Document JVT-I017, 2003.

[36] K. Sayood, "Introduction to data compression," USA: Morgan Kaufmann Publishers, 2005.

[37] C. Poynton, "Digital Video and HD: Algorithms and Interfaces," USA: Morgan Kaufmann Publishers, 2003.

[38] P. Roos, M. A. Viergever, M. C. A. van Dijke, and J. H. Peters, "Reversible intraframe compression of medical images," IEEE Trans. Med. Imag., vol. 7, no. 4, pp. 328-336, 1988.

[39] W. Sweldens, "The lifting scheme: a custom-design construction of biorthogonal wavelets," Applied and Computational Harmonic Analysis, vol. 3, pp. 186-200, 1996.

[40] R. Ramanath, S. Wesley, Y. Yoo, and M. Drew. "Color image processing pipeline," IEEE Signal Processing Magazine, vol. 22, no. 1, pp. 34-43, Mar. 2005.

[41] N. X. Lian, L. Chang, V. Zagorodnov, and Y. P. Tan, "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.

[42] P. Roos, M.A. Viergever, M.C.A. van Dijke, and J.H. Peters, "Reversible intraframe compression of medical images," IEEE Trans. on Medical Imaging, vol. 7, no. 4, pp. 328-336, Dec. 1988 .

[43] I. Daubechies and W. Sweldens, "Factoring wavelet transforms into lifting steps," Journal of Fourier analysis and applications, vol. 4, no. 3, pp. 247-269, 1998.

[44] C. Doutre and P. Nasiopoulos, "Modified H. 264 intra prediction for compression of video and images captured with a color filter array," IEEE International Conference on Image Processing (ICIP), pp. 3401-3404. Nov. 2009.

[45] H. Chen, M. Sun, and E. Steinbach, "Compression of Bayer-pattern video sequences using adjusted chroma subsampling," IEEE Trans. on Circuits and Systems for Video Technology, vol. 19, no. 12, pp. 1891-1896, Sep. 2009.

[46] X. Wu, G. Zhai, X. Yang, and W. Zhang, "Adaptive sequential prediction of multidimensional signals with applications to lossless image coding," IEEE Trans. on Image Process., vol. 20, no. 1, pp. 36-42, Jan. 2011.

초록

무손실 압축은 메모리와 대역폭을 많이 사용하기 때문에 손실 압축에 비해 덜 사용된다. 하지만 의료, 출판, 과학, 예술과 같은 분야에서는 무손실 압축이 필수적이다. 또한 카메라와 영상 표시 장치의 성능이 향상되고 메모리의 가격은 낮아지면서 압축으로 인한 영상 열화를 피하고자하는 요구가 증가하고 있다. 따라서 많은 경우에 손실 압축이 사용됨에도 불구하고 효율적인 무손실 압축의 중요성은 점점 커지고 있다. 본 논문에서는 채널 간의 상관관계를 이용한 컬러 영상과 컬러 필터 어레이 영상의 무손실 압축 방법을 제안한다.

먼저 기존의 무손실 색상 변환 방법에 lifting step을 추가하여 성능을 향상시킨 새로운 무손실 색상 변환 방법을 제안한다. 이미지의 적색, 녹색, 청색 채널 간에는 높은 수준의 상관관계가 있는데, 이는 YCbCr 변환을 통해서 성공적으로 제거된다. 하지만, 이 변환은 무손실 압축에는 사용될 수 없기 때문에 JPEG2000을 포함한 표준 무손실 압축 방법에서는 무손실 색상 변환이 사용된다. 이 변환은 가역적이고 간단하기 때문에 성능이 부족하여, 이를 향상시킬 필요가 있다. 간단하지만 효과적인 연산을 통해서 제안하는 방법은 기존의 무손실 색상 변환의 성능을 향상시켜 YCbCr과 비슷한 성능을 보여준다. 또한, 표준 압축 방법인 JPEG-LS를 이용하여 실험한 결과 기존의 무손실 색상 변환에 비해 1.46%의 압축률 향상 효과가 있었다.

다음으로 계층적인 예측을 통한 컬러 영상의 무손실 압축 방법을 제안한다. 앞에서 제안한 새로운 무손실 색상 변환을 통해 RGB 영상을 변환하고, 밝기 성분인

Y는 기존의 무손실 압축 방법으로 압축을 한다. 색상 채널 C'_u와 C'_v는 제안하는 계층적으로 분해되어 방향성을 이용해 예측되며, 예측 오차는 context modeling을 통해 부호화된다. 제안하는 방법은 Kodak 영상, 의료 영상, 디지털 카메라 영상에 대해 여러 기존 방법과 비교되었으며, 가장 우수한 JPEG2000 대비 각 셋에 대해 5.85%, 10.40%, 4.89% 향상된 압축률을 얻었다.

마지막으로 효과적인 context modeling을 이용한 Bayer 컬러 필터 어레이 영상의 무손실 압축 방법을 제안한다. 예측 오차의 효과적인 context modeling을 위해서 계층적 예측이 사용되었으며, 이 과정에서 mosaic 영상은 4개의 부영상으로 나누어지고 순서대로 압축된다. 임의의 부영상을 예측하기 위해서 이전에 압축된 부영상이 모두 이용되어 해당 픽셀 위치의 에지 방향과 후보 예측값을 계산한다. 이 값들은 다시 예측 오차의 효과적인 context modeling을 위해서도 사용된다. 제안한 방법은 실제 컬러 필터 어레이 영상과 Kodak 영상, 디지털 카메라 영상에 대한 실험에서 비교한 모든 방법보다 우수한 성능을 보였다.

주요어: 무손실 컬러 영상 압축, 무손실 색상 변환, 리프팅, 계층적 압축, 컬러 필터 어레이 압축

학 번: 2007-20944