A LOW COMPLEXITY IMAGE COMPRESSION ALGORITHM FOR BAYER COLOR FILTER ARRAY

A Thesis Submitted to the College of Graduate Postdoctoral Studies
In Partial Fulfillment of the Requirements
For the Degree of Master of Science
In the Department of Electrical and Computer Engineering
University of Saskatchewan
Saskatoon

By

K M Mafijur Rahman

PERMISSION TO USE

In presenting this thesis in partial fulfillment of the requirements for a Postgraduate degree from the University of Saskatchewan, I agree that the Libraries of this University may make it freely available for inspection. I further agree that permission for copying of this thesis in any manner, in whole or in part, for scholarly purposes may be granted by the professor or professors who supervised my thesis work or, in their absence, by the Head of the Department or the Dean of the College in which my thesis work was done. It is understood that any copying or publication or use of this thesis or parts thereof for financial gain shall not be allowed without my written permission. It is also understood that due recognition shall be given to me and to the University of Saskatchewan in any scholarly use which may be made of any material in my thesis/dissertation.

Requests for permission to copy or to make other use of material in this thesis in whole or part should be addressed to:

Head of the Department of Electrical and Computer Engineering
57 Campus Drive
University of Saskatchewan
Saskatoon, Saskatchewan, S7N 5A9,
Canada

OR

Dean
College of Graduate and Postdoctoral Studies
University of Saskatchewan
116 Thorvaldson Building, 110 Science Place
Saskatoon, Saskatchewan S7N 5C9,
Canada

ABSTRACT

Digital image in their raw form requires an excessive amount of storage capacity. Image compression is a process of reducing the cost of storage and transmission of image data. The compression algorithm reduces the file size so that it requires less storage or transmission bandwidth. This work presents a new color transformation and compression algorithm for the Bayer color filter array (CFA) images. In a full color image, each pixel contains R, G, and B components. A CFA image contains single channel information in each pixel position, demosaicking is required to construct a full color image. For each pixel, demosaicking constructs the missing two-color information by using information from neighbouring pixels. After demosaicking, each pixel contains R, G, and B information, and a full color image is constructed. Conventional CFA compression occurs after the demosaicking. However, the Bayer CFA image can be compressed before demosaicking which is called compression-first method, and the algorithm proposed in this research follows the compression-first or direct compression method. The compression-first method applies the compression algorithm directly onto the CFA data and shifts demosaicking to the other end of the transmission and storage process. The advantage of the compression-first method is that it requires three time less transmission bandwidth for each pixel than conventional compression.

Compression-first method of CFA data produces spatial redundancy, artifacts, and false high frequencies. The process requires a color transformation with less correlation among the color components than that Bayer RGB color space. This work analyzes correlation coefficient, standard deviation, entropy, and intensity range of the Bayer RGB color components. The analysis provides two efficient color transformations in terms of features of color transformation. The proposed color components show lesser correlation coefficient than occurs with the Bayer RGB color components. Color transformations reduce both the spatial and spectral redundancies of the Bayer CFA image. After color transformation, the components are independently encoded using differential pulse-code modulation (DPCM) in raster order fashion. The residue error of DPCM is mapped to a positive integer for the adaptive Golomb rice code. The compression algorithm includes both the adaptive Golomb rice and Unary coding to generate bit stream. Extensive simulation analysis is performed on both simulated CFA and real CFA datasets. This analysis is extended for the WCE (wireless capsule endoscopic) images. The compression algorithm is also realized with a simulated

WCE CFA dataset. The results show that the proposed algorithm requires less bits per pixel than the conventional CFA compression. The algorithm also outperforms recent works on CFA compression algorithms for both real and simulated CFA datasets.

ACKNOWLEDGEMENTS

I would like to start by praising the almighty God, Allah, the creator of the havens and the earth. Special thanks to my supervisor, Professor Khan A. Wahid. In his supervision and proper guidance, I was able to acquire complete knowledge and skill in image compression algorithm and their implementation. His encouragement for hard-work always motivates me in the field of learning.

I would like to thank Professor Anh Dinh, Professor Francis Bui, Professor Seok-Bum Ko and Professor Khan A. Wahid for giving me the opportunity to take excellent courses at the University of Saskatchewan. I would also like to thank Shahed Khan Mohammed for the inspiration and knowledge sharing.

Finally, I would like to thank my family for their love, encouragement and support.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABBREVIATIONS AND SYMBOLS

BPP	Bits per pixel
CFA	Color filter array
CMOS	Complementary metal oxide semiconductor
CPSNR	Color peak signal to noise ratio
CR	Compression ratio
CT	Color transformation
DCT	Discrete cosine transforms
DWT	Discrete wavelet transforms
DPCM	Delta pulse modulation
FPGA	Field programmable gate array
HDL	Hardware description language
HP	Hierarchical prediction
HVS	Human Visual System
JPEG	Joint Photographic expert group
MEP	Median edge prediction
MSE	Mean square error
RGB	Red green blue
SSIM	Structural similarity

Chapter 1 INTRODUCTION

In this era of the Internet, the transmission and storage of image data is a necessary part of technology. Image data is widely used in every aspect of modern life such as social media, communication, advertising, remote diagnosis and analysis, medical imaging, and security surveillance. Considering the huge number of image applications, the corresponding raw data requires huge storage capacity. For images to be stored or transmitted, they must be compressed. Image compression reduces the size of an image, providing the exact image or degraded version of the image as needed and lessening the required bandwidth, reducing transmission time, decreasing network congestion, and lowering costs. The most widely used method for image compression is shown as block diagram in Figure 1-1.

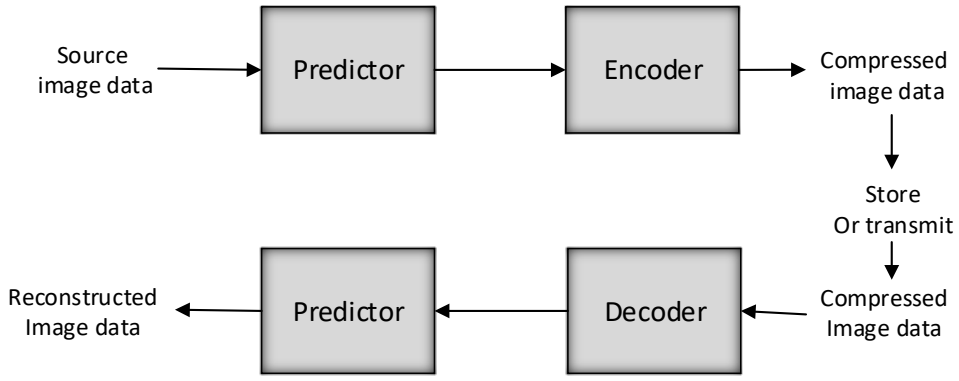


Figure 1-1: General block diagram of an image compression

A popular low-cost method in digital image processing is the Bayer CFA image, which has a reputation for decreasing transmission bandwidth, reducing storage, and saving energy. Bayer CFA is a single sensor digital image, which contains RGB information in a Bayer pattern, as shown in Figure 1-2 [1]. A Bayer filter mosaic is a color filter array for arranging RGB color filters on a square grid of photo sensors. The specific pattern of color filters is used in most of the single-chip digital image sensors to create a color image. The Bayer color filter pattern contains 50% green, 25% red and 25% blue. Therefore, the Bayer pattern has two green (G_r, G_b) pixels, one red (R) pixel and one blue (B) pixel in a 2 x 2 square block which is also called RG_rBG_b, G_rRG_bB, or RG_rG_bB. The Bayer array measures the G image on a quincunx grid and the R & B images on rectangular grids. The G image is measured at a higher sampling rate because the human eyelid is

sensitive to medium wavelengths that, correspond to the G portion of the spectrum. In each pixel position, there is single-channel information; therefore, information for the other two channels are interpolated from the neighboring pixel position.

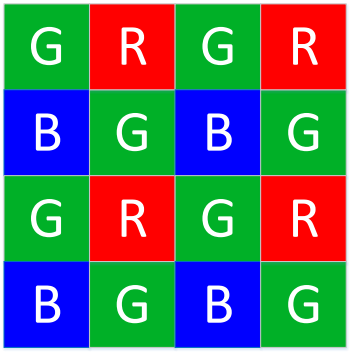


Figure 1-2 Bayer Color Filter Array pattern

A full color image is generated from the captured data by demosaicking method. The generated color image is compressed using different compression algorithms, such as JPEG, JPEG-LS, or JPEG2000, before storage or transmission. The compression algorithm can be applied before the demosaicking step. This method is more popular than the conventional compression method.

1.1 Research background

Since the discovery of the CFA image, different studies have been conducted on the design of the CFA compression algorithm. This section discusses previous works on the design of this algorithm. The compression in RGB color space of a CFA image has spectral redundancy, spatial redundancy, artifacts, and false high frequencies [3] – [10]. Koh [4] shows that Bayer RGB color components are not suitable for compression-first method. Therefore, the focus of the current work is on the color transformation of the CFA image.

In CFA image compression, compression is usually carried out after the demosaicking process. Therefore, three color components are realized with the compression algorithm which needs more transmission bandwidth and processing cost. The compression can also be performed before the demosaicking process [3] – [11]. This compression-first method saves two-thirds of transmission

bandwidth. Although the compression-first (direct compression) of CFA data is efficient, it has limitations with existing encoding methods. Koh [4] shows that the compression-first method of CFA data with JPEG introduces poor quality images. In [4], composite peak signal to noise ratio (CPSNR) defined as Eqn. 1.1 is used to measure the quality of the image.

$$CPSNR = 10\log_2\left(\frac{255^2}{\frac{1}{3MN}\sum_{\forall k}\left[I_{in}(i,j,k) - I_{out}(i,j,k)\right]^2}\right) \tag{1.1}$$

Here, I_{in} and I_{out} are the input and output images respectively. M and N are the dimension of each color channel, i and j are the locations of the pixel in color plane k.

The value of CPSNR for the conventional compression method is 33.37~db, and for directly compressed CFA data using JPEG, it is 30.04~db. The image is distorted in the compression-first method due to the high correlation among the three-color planes. The high frequency data in the image, does not allow high compression ratio for JPEG compression. This result demonstrates that the Bayer-patterned CFA data is not suitable for compression-first method using JPEG. To resolve this problem with the compression-first method of CFA data using the JPEG, Koh [4] and Ortega [10] proposed three methods: structure rearranging, filtering, and conversion. The rearranging structure method is differently proposed in [4] and [10]. This method merges the odd and even columns of the image. As all the even columns are shifted one pixel to the left and all the zero columns removed, false high frequencies are created in the reconstructed image. To reduce the false high frequencies generated by merging the two columns, [11] explains that the quincunx grid can be visualized as two interlaced frames of a scene and that by de-interlacing these frames, a smoother image is obtained. The structure separation method separates the quincunx array into two arrays. Another approach is to use the low-pass filter to process CFA data. As the CFA image contains more high frequency components than a full color image, therefore the compression produces artifacts. The high frequencies can be removed by applying low-pass filters. A third approach is conversion. Koh [4] uses the color conversion for the CFA data to reduce the artifacts. The structure and filter methods are applied after the conversion of the RGB to a different color space. Koh applied the color conversion by considering a 2 x 2 block of a Bayer pattern into a 2 x 2 of $Y_{ul}Y_{lr}C_bC_r$. He represented the color conversion matrix from RGB to YCbCr as shown in Eqn.

1.2. The conversion method is used with the structure or low-pass filter methods. Ortega, Toi and Koh used conversion rather than CFA RGB color space for the compression algorithm. The conversion is an efficient step in the Bayer CFA compression algorithm.

$$\begin{bmatrix} Y_{ul} \\ Y_{lr} \\ C_b \\ C_r \end{bmatrix} = \begin{bmatrix} 128.6 & 0 & 25 & 65.5 \\ 0 & 128.6 & 25 & 65.5 \\ -37.1 & -37.1 & 1/8 & -37.8 \\ -46.9 & -46.9 & -18.2 & 112 \end{bmatrix} \begin{bmatrix} G_{ul} \\ G_{lr} \\ B \\ R \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 128 \\ 128 \end{bmatrix} \quad (1.2)$$

An efficient color transformation is a necessary step for CFA compression algorithm. In the current work, two different new CFA color conversions are proposed for the compression algorithm and new color transformations are proposed by analyzing the features of the RGB color components. Wireless capsule endospy (WCE) images shows better performance with these features of color transormation [25][26]. The previous research discusses the wireless capsule endoscpy (WCE) compression for the Bayer CFA image [13] – [14]. The YLMN compression algorithm uses compression-first method and a new YLMN color transformation [13]. The correlation among the color components is relatively high in [13]. Therefore, the compression performance needs to be improved to save the transmission bandwidth and power for the WCE system.

1.2 Design challenges

This work focuses on the design of an efficient compression algorithm that has limited bandwidth, is low power, is low cost, and shows good performance. The design of the proposed efficient compression algorithm has experienced several challenges, including the compression-first method of the CFA image, image compatibility, and the implementation of the design.

1.2.1 Direct compression of a CFA image

The compression-first method of a CFA image is a widely used technique to save power and transmission bandwidth in image processing. The compression is applied after the demosaicking of the CFA image in the conventional method. As discussed earlier, it requires higher storage or transmission bandwidth and power. Therefore, the compression algorithm shifts the demosaicking stage to after transmission or storage. This compression algorithm is simpler than conventional

algorithm. It is efficient with bandwidth and removes the high computation of demosaicking. Compression-first method is not efficient with the JPEG or JPEG-2000 compression algorithm. As discussed earlier, it causes redundancies, artifacts, and discontinuities. The proposed work uses color transformation to resolve these issues.

1.2.2 Design implementation

The design implementation depends on memory usage, computational complexity and power. The computational complexity of the proposed work is simple and leads to low power consumption. It requires color transformation, a DPCM prediction model and adaptive Golomb rice coding. The design has the potential to be easily integrated with the existing image sensor.

1.3 Thesis objectives

The purpose of this thesis is to design an efficient image compression algorithm that works for both natural and WCE Bayer color filter array images. The major objectives are as follows:

1. To improve the existing color transformation methods in CFA image compression for reducing the redundancy between different color components in CFA image and increasing the compression ratio.
2. To develop a compression algorithm for both natural images and WCE images for a better compression performance for Bayer CFA.
3. To improve the power and memory usage in the CFA compression algorithm with a simple color transformation, using only shifter and adder, the DPCM prediction model, adaptive Golomb rice coding, and unary coding. Hence, it reduces implementation cost.

1.4 Thesis outline

The thesis comprises six chapters. Chapter 2 provides an overview of image compression for the Bayer color filter array image by classifying types of image compression, presenting different color transformations, and describing different types of encoding and prediction models. Chapter 3 describes in detail, the proposed compression design. Chapter 4 presents the performance of the compression design for different datasets. Chapter 5 describes the hardware design. Chapter 6 summarizes the work's accomplishment and provides recommendations for the future.

Chapter 2 OVERVIEW OF AN IMAGE COMPRESSION

The design of an image compression algorithm is required to minimize the storage and transmission bandwidth cost. It reduces the processing power and transmission power on the transmission line. Therefore, image and video transmission on internet and server become significantly popular. It can reduce the image size without the significant degradation of the reconstructed image quality or with no degradation of reconstructed image. In this work, the design is focused on the efficient image compression for the power limited application. Wireless capsule endoscopy is one of the achievable applications.

The basic technique of the compression is human visual system, which is sensitive to a limited bandwidth. The image sensor captures wide range of bandwidth. Therefore, it contains lot of redundancy. Image compression eliminates these redundancies and reduces the image size. Based on the process of removing redundancy, image compression can be classified into two categories, lossless and lossy compression. In this work, lossless image compression is designed to achieve low power and low complexity compression.

2.1 Lossless compression

Digital images contain lots of redundant information; thus, compression algorithm removes the redundancy and minimizes the storage space. If the compression algorithm is reversible such as it can reconstruct the original image, then it is called lossless image compression [15]. The lossless compression is mostly focused on de-correlation and entropy coding. De-correlation removes the spatial redundancy or inter-pixel redundancy. On the other hand, entropy coding removes the coding redundancy. Huffman coding, Arithmetic coding, and LWZ are widely used encoding method in lossless compression algorithm. Lossless compression is mostly used in high performance applications such as geophysics, telemetry, medical science etc.

2.2 Lossy compression

The image compression removes the redundant information; if the reconstructed image is not exactly the original image then it is called a lossy compression. The lossy compression is not reversible compression rather it loses image information, which cannot be recovered by the reverse process. Reconstruction of image approximates original image, therefore the quality measurement

of the image for the lossy compression is necessary. It degrades the image quality after compression. It provides with high compression ratio and often applied in video and image compression. When the quality of the image is not readily necessary rather then transmission bandwidth, a lossy compression is more suitable. Often it is applied for the surveillance camera for security purpose.

2.3 Overview of Bayer CFA

In most imaging systems, image compression is carried out after the demosaicking process. The compression algorithms are focused on three channel color images. Figure 2-1 shows the overall diagram of most widely used imaging system. When the compressed images are transmitted or stored from the image captured module, the connection between the two systems is bandwidth limited. This is the bottleneck of whole system performance. Therefore, efficient image compression methods are exercised to send less amount of data to the link and they are commonly applied after the demosaick operation [15].

In several previous works, alternative image compression algorithm is proposed as shown in Figure 2-2 [4] - [13].

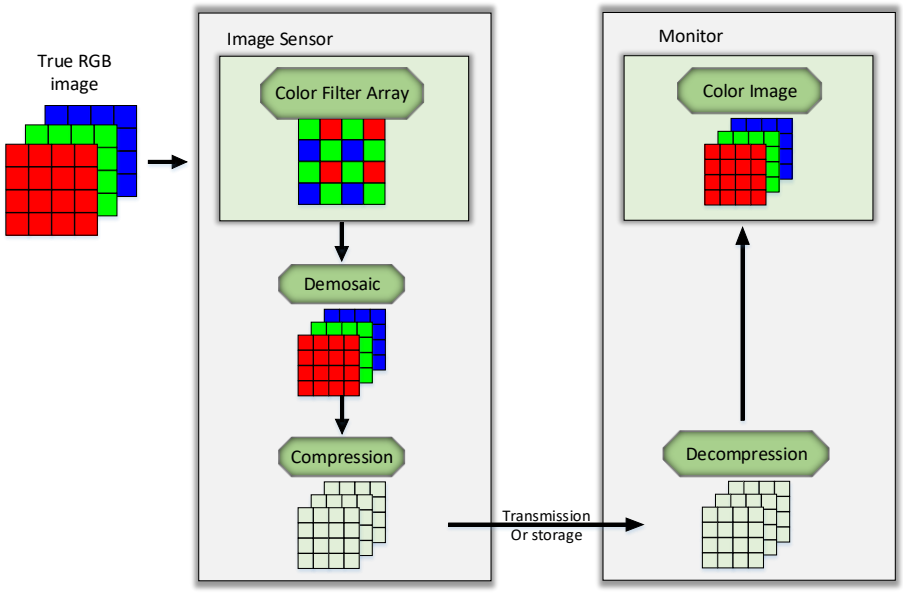


Figure 2-1: Illustration of digital color imaging systems for conventional CFA image compression.

The key difference of this system is that it directly compresses CFA image and the demosaicking operation is completed after the decompression of the image data. By compressing the CFA data, the compression algorithm deals with only one third of the full image data. In conventional compression, three channel RGB images needs to be compressed where the demosaicking enables the redundancy of the image data. In this compression-first method, it does not have to deal with the redundancy information that are introduced with the color demosaicking, which provides with compression that is more efficient.

In addition, CFA compression scheme reduces the computational burden to the decoder of the system. The decoder consumes more computational power and memory, which is being reduced by this CFA compression scheme.

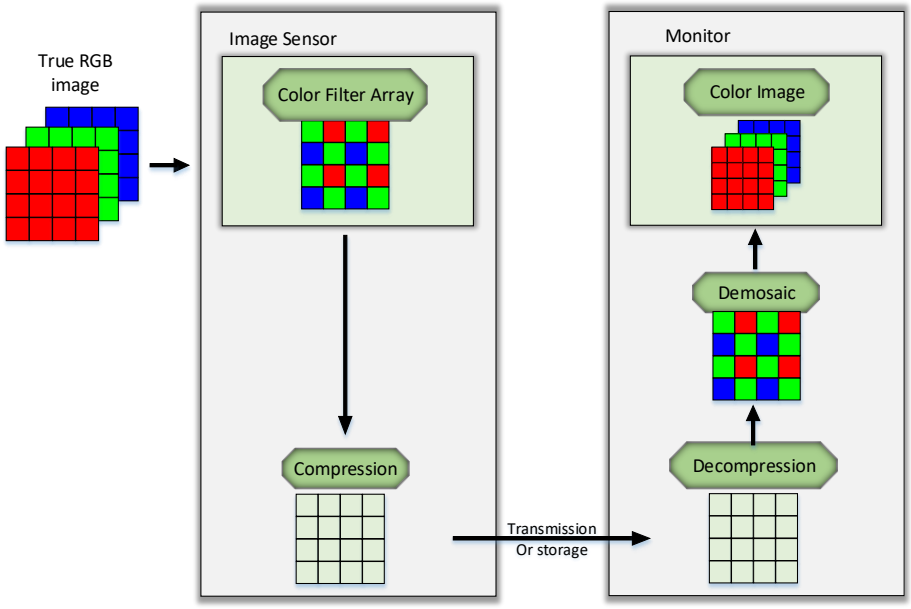


Figure 2-2 Illustration of digital color imaging systems for alternate CFA image compression.

Many devices work in real time; therefore, those systems have very strong constraints in power and processing time. This alternate CFA compression scheme passes the whole de-mosaic stage to the other side and saves the power and processing time. This proposed pipeline has great advantage in minimizing the workload on power-limited image capturing module.

2.4 Color transformation

Color transformation is a most important part of compression design in Bayer CFA image. In the compression design, color transformation can be lossless or lossy which also determines the type of compression. The color transformation does not compress the image directly rather it enhances the compression ratio by removing the redundancy among the color components. It separates the chrominance and luminance information of an image. Therefore, it enhances the prediction model and improves the residues error of the prediction model. To get further savings on the bandwidth, chrominance components could be subsampled to remove the spectral redundancy and it will become a lossy compression. In lossless compression, the quality of image should be same as the original therefore subsampling is not included in the lossless compression design. The standard color space of the Bayer image is G_rRBG_b. It is the most popular energy saving technique in digital image processing. It contains RGB information in Bayer pattern. The pattern contains two green pixels (G), one red (R) and one blue pixel (B). Therefore, the correlation among the color components in Bayer G_rRBG_b is high which reduces the performance of the compression design. The color transformation ensures the less correlation among the components. Therefore, the color transformation becomes the essential part of the compression design. There are several existing color transformations available in the literature. Each of them contains different usages and significance.

2.4.1 Color transform: RGB

Color machine vision systems generally capture images in the red, green, blue (RGB) color system as 24-bit images. Each color is allocated 8 bits, resulting 256 different values. RGB can be thought of as three grayscale images representing the light values of Red, Green, and Blue. Most graphics applications work in RGB. In case of compression, RGB color components contain spatial redundancy. Correlation coefficient is high in RGB color transformation. Therefore, RGB color transformation has a low compression ratio [55].

(a) RGB (b) R (c) G (d) B

Figure 2-3: RGB color image in (a) and RGB color components R, G, and B in (b), (c), and (d) respectively

2.4.2 Color transform: YC_bC_r

YC_bC_r color transform represents the human perception of color more closely than the standard RGB model used in computer graphics. Y is the luminance (brightness) component and C_b and C_r are the chrominance (color) components in YC_bC_r color transformation. The correlation among the YCbCr color components are lower than the RGB color space. It reduces the spatial redundancy in the image. It performs better in image compression [55].

(a) YC_bC_r (b) Y (c) C_b (d) C_r

Figure 2-4: YC_bC_r color image (a) and Y, C_b, and C_r components in (b), (c), and (d) respectively.

2.4.3 Color transform: Bayer G_rRBG_b

Bayer color filter array is a popular format for digital acquisition of color images [1]. The half of the total number of pixels are green, while a quarter of the total number of pixels is assigned to both red and blue. The reason behind the 50% of green pixels is the sensitivity of the human eye. Human eye is sensitive to light of a certain intensity range of wavelength. The average normal

sighted of human eye is most sensitive at wavelength of 555 nm; resulting in the fact that green light at this wavelength produces the impression of highest “brightness” compared to the other wavelengths, which is shown in Figure 2-5. Therefore, the green color has more contribution to the human eye rather than other color components. The popular Bayer pattern 2x2 block contains

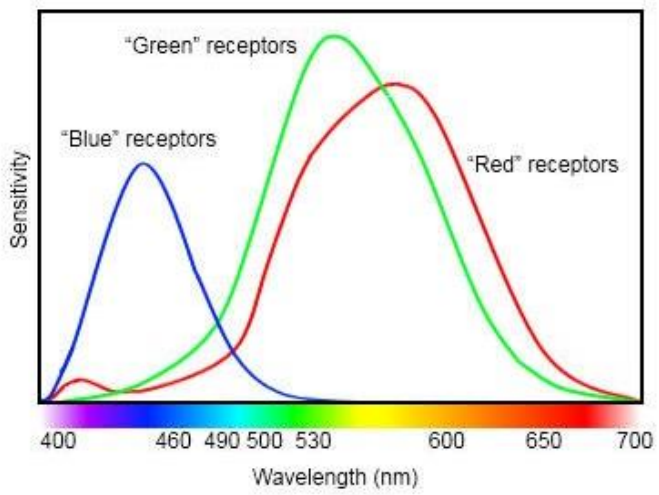


Figure 2-5: Human Color receptor relative sensitivity (Photograph adapted from Photobit)

two green pixels, a blue and a red pixel. Three filter uses to capture the color information from the sensor. The color information can be obtained by using the color image sensor, which is covered with either a red, a green or a blue filter in a repeating pattern. This pattern of filters can vary but the widely adopted “Bayer” pattern is a repeating 2x2 arrangement. The pixel sequence of the image sensor comes out GRGRGR, etc., and then the other line is BGBGBG, etc. The Figure 2-6 shows how the pixels are captured one by one. This is called the sequential RGB or sRGB. As each contains only one-color information therefore, the overall sensitivity of the color image sensor is less sensitive. In the other way around, it saves two third of pixel information.

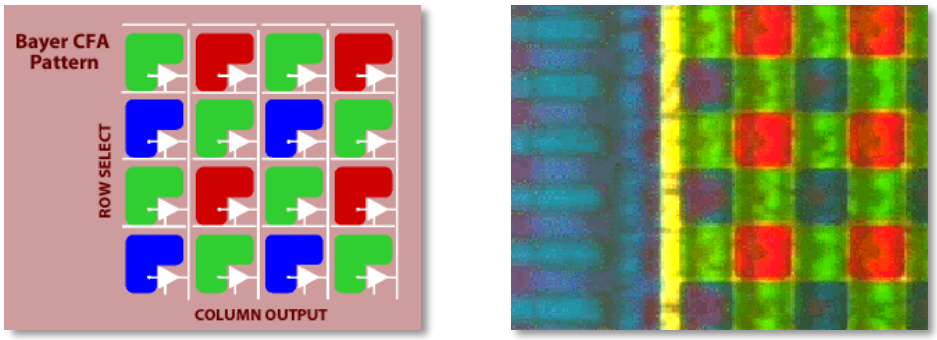


Figure 2-6: Image sensor capturing Bayer filter image pixel in raster order fashion (Photographs adapted from Photobit)

(a) B

(b) G_b

(c) G_r

(d) R

Figure 2-7: Bayer color filter array image G_rRBG_b components

There are several established demosaicking methods available to get the full color image from Bayer image. Each pixel generates other two-color information from their neighbor's color information. The full color image is created from the demosaicking.

2.4.4 Color transform: Y_1Y_2CbCr

CFA images consist with four-color components. YCbCr color plane has three components [19]. The components are classified with two classes such as luminance component and chrominance component. The luminance information gives the grayscale image of the color image. On the other hand, the chrominance components hold the color information. Therefore, the Y component is sufficient to represent the black and white image, which is used in black and white television, and three components produce the color image, which is used in color television. There are several ways of converting Bayer RGB to YCbCr and the popular conversion of the YCbCr is that Y_1 and Y_2 can be represented with its samples on the quincunx grid where G_r and G_b are sampled respectively, C_b and C_r can be represented with samples on the interlacing grids respectively where B and R are sampled [11].

To convert the Bayer RGB to Y_1Y_2CbCr color transformation, following Eqn. 2.1 is used.

$$\begin{bmatrix} Y_1 \\ Y_2 \\ C_b \\ C_r \end{bmatrix} = \begin{bmatrix} 1/2 & 1/4 & 1/4 & 0 \\ 0 & 1/4 & 1/4 & 1/2 \\ -1/4 & 0 & 1/2 & -1/4 \\ -1/4 & 1/2 & 0 & -1/4 \end{bmatrix} \begin{bmatrix} Gr \\ R \\ B \\ Gb \end{bmatrix} \tag{2.1}$$

Where R, G, B are the red, green and blue components of Bayer RGB plane.

To get back the Bayer RGB components from the Y_1Y_2CbCr transformation, following Eqn. 2.2 is used.

$$\begin{bmatrix} Gr \\ R \\ B \\ Gb \end{bmatrix} = \begin{bmatrix} 3/2 & -1/2 & -1/2 & -1/2 \\ 1/2 & 1/2 & -1/2 & 3/2 \\ 1/2 & 1/2 & 3/2 & -1/2 \\ -1/2 & 3/2 & -1/2 & -1/2 \end{bmatrix} \begin{bmatrix} Y_1 \\ Y_2 \\ C_b \\ C_r \end{bmatrix} \quad (2.2)$$

The color components Y1, Y2, Cb, and Cr are shown in Figure 2-8.

(a) Y_1

(b) C_b

(c) C_b

(d) Y_2

Figure 2-8: Color components of Y_1 $Y_2C_bC_r$ for Bayer CFA image

2.4.5 Color transform: $YD_gC_oC_g$

A new color space for Bayer images is described in [16]. The $YD_gC_oC_g$ color plane is similar to the $Y_1Y_2C_bC_r$ as it is based on the luminance and chrominance. Y is the only luminance

components and all the other components D_g, C_o, C_g carry the chrominance information of the Bayer image. The Y channel is an average of four components of the macro-pixel which contains

$$\begin{bmatrix} Y \\ D_g \\ C_o \\ C_g \end{bmatrix} = \begin{bmatrix} 1/4 & 1/4 & 1/4 & 1/4 \\ -1 & 1 & 0 & 0 \\ 0 & 0 & 1 & -1 \\ 1/2 & 1/2 & -1/2 & -1/2 \end{bmatrix} \begin{bmatrix} G_r \\ G_b \\ R \\ B \end{bmatrix} \tag{2.3}$$

$$\begin{bmatrix} G_r \\ G_b \\ R \\ B \end{bmatrix} = \begin{bmatrix} 1 & -1/2 & 0 & 1/2 \\ 1 & 1/2 & 0 & 1/2 \\ 1 & 0 & 1/2 & -1/2 \\ 1 & 0 & -1/2 & -1/2 \end{bmatrix} \begin{bmatrix} Y \\ D_g \\ C_o \\ C_g \end{bmatrix} \tag{2.4}$$

(a) Y

(b) D_g

(c) C_o

(d) C_g

Figure 2-9: $YD_gC_oC_g$ color components for Bayer CFA image

50% green and 25% contribution of R and B. As the all other components contain chrominance information therefore, the gray level image has the value of $D_g = C_o = C_g = 0$. D_g is the difference of green pixels and C_g is the excess green channel where the C_o carries the orange channel information. The direct and inverse transform matrices have only the value of 0, ¼, ½, or 1. Therefore, it has the less computational complexity and which uses the right shift operator only for the implementation. The Eqn. 2.3 and 2.4 are representing the direct and inverse transform of the $YD_gC_oC_g$ color plane respectively. The significance of each color components is shown in Figure 2-9.

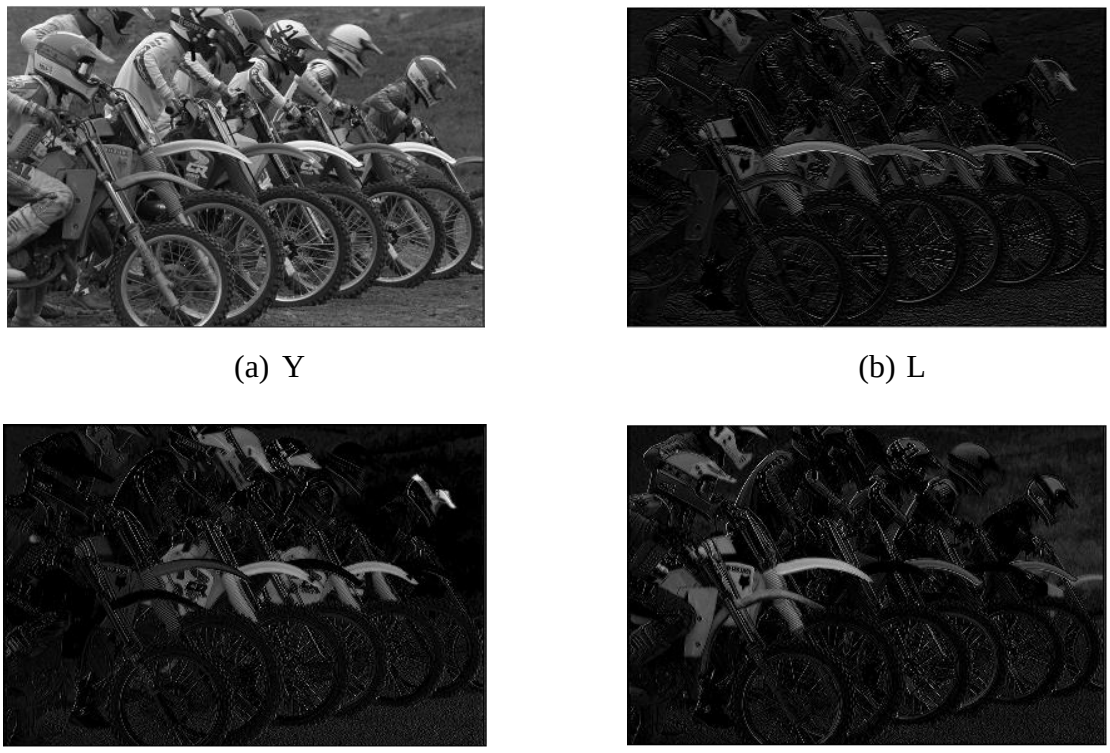
2.4.6 Color transform: YLMN

The YLMN color transformation in Bayer CFA compression is efficient compare to Bayer RGB color transformation. In [13], YLMN is derived to exploit the inter-color correlation in CFA image. The YLMN color transform is lossless and it works directly in raster order fashion, which is efficient in implementation. The color transformation described mathematically in Eqn. 2.5 and 2.6.

$$\begin{bmatrix} Y \\ L \\ M \\ N \end{bmatrix} = \begin{bmatrix} 1/4 & 1/4 & 1/4 & 1/4 \\ 1/2 & 1/2 & -1/2 & -1/2 \\ -1 & 1 & 0 & 0 \\ 0 & 0 & -1 & 1 \end{bmatrix} \begin{bmatrix} G_r \\ R \\ B \\ G_b \end{bmatrix} \tag{2.5}$$

$$\begin{bmatrix} G_r \\ R \\ B \\ G_b \end{bmatrix} = \begin{bmatrix} 1 & 1/2 & -1/2 & 0 \\ 1 & 1/2 & -1/2 & 0 \\ 1 & -1/2 & 0 & -1/2 \\ 1 & -1/2 & 0 & 1/2 \end{bmatrix} \begin{bmatrix} Y \\ L \\ M \\ N \end{bmatrix} \tag{2.6}$$

The color components of the YLMN are shown in Figure 2-10.



(a) Y (b) L

(c) M (d) N

Figure 2-10: YLMN color components

2.5 Feature parameter of color transformation

The color transformation performance depends on different parameter. Entropy, standard deviation, correlation and prediction gain are widely used feature parameter of color transformation. The definition of these parameters is discussed in this sub section.

2.5.1 Entropy

Image entropy is a quantity, which is measured to explain the business of an image. It also gives the amount of information, which must be coded by a compression algorithm. Low entropy image contains low contrast and a large number of similar intensity pixels. Consequently, it can be compressed to relatively small size. On the other hand, high contrast and a great deal of variation in intensity value from one pixel to another provide high entropy. Therefore, it cannot be compressed as much as low entropy images. The image entropy is calculated in the proposed algorithm using the Eqn. 2.7.

$$Entropy = -\sum_i P_i Log_2 P_i \qquad (2.7)$$

In the above expression, Pi is the probability that the difference between 2 adjacent pixels is equal to i.

2.5.2 Standard deviation

Standard deviation is a measure of the dispersion of a set of data from its mean. It is calculated as the square root of variance by determining the variation between each data point relative to the mean. If the data points are further from the mean, there is higher deviation within the dataset. The mathematical definition of standard deviation can be described as Eqn. 2.8.

$$SD = \sqrt{\frac{\sum |x - \mu|^2}{N}} \qquad (2.8)$$

Where x is the intensity value of the image pixel, μ is the mean of image data set and N is the number total pixels in the image.

2.6 Transform coding

Transform coding is block based image data processing with an efficient de-correlation technique. Transform coding, first transforms the image from its special domain representation to a different type of representation using the existing transform methods, which produce transformed co-efficient. The widely used transform methods are Discrete Cosine Transform (DCT) and Discrete Wavelet Transform (DWT).

2.6.1 Discrete cosine transform (DCT)

An image transforms from the discrete space domain to the discrete spatial frequency domain is called DCT transform [24]. DCT is orthogonal transform and it contains high-energy compacting capability. The image data is condensed into a relatively small number of transformed coefficient.

2.6.2 Discrete wavelet transform (DWT)

The Discrete Wavelet Transform (DWT) is a very versatile signal-processing tool after Mallat proposed the multi-resolution representation of signals based on wavelet decomposition. Wavelets allow both time and frequency analysis of signals simultaneously because the energy of wavelets is concentrated in time and still possesses the wave-like (periodic) characteristics. As a result, wavelet representation provides a versatile mathematical tool to analyze transient, time-variant (non-stationary) signals that are not statistically predictable especially at the region of discontinuities – a feature that is typical of images having discontinuities at the edges. The DWT decomposes a digital signal into different sub-bands so that the lower frequency sub-bands have finer frequency resolution and coarser time resolution compared to the higher frequency sub-bands. DWT is the basis of the new JPEG2000 image compression standard. DWT is computationally expensive to implement.

2.7 Line coding

The most common line coding is differential pulse coded modulation (DPCM). In DPCM, the current signal information is deducted from the previous signal information to create the residue error. Then the residue error is sent to the encoder to generate compressed bit stream. Line coding is computationally simple. It gives lossless compression and implementation cost is low. Therefore, low complexity and low-cost design uses line coding rather than transform coding.

2.8 Image coder

2.8.1 Source encoder

Source encoder efficiently processes the output of digital source into a sequence of binary digits. A variety of linear transformations are developed using Discrete Cosine Transformation (DCT) and Discrete Wavelet Transformation (DWT). The encoder uses Discrete Fourier Transformation (DFT), Discrete Cosine Transformation (DCT) or Discrete Wavelet Transformation (DWT). Each of them has several advantages and disadvantages.

2.8.2 Entropy encoder

An entropy encoder further compresses the values lossless to give better overall compression. It uses a model to accurately determine the probabilities for each value and produces an appropriate code based on these probabilities so that the resultant output code stream will be smaller than the input stream. The most commonly used entropy encoders are the Huffman encoder and the arithmetic encoder, although for applications requiring fast execution, simple run-length encoding (RLE) has shown very effective. The encoder depends on the range of the data. If the range of the data is small, then Unary coding, Golomb coding, Rice coding, Golomb rice coding and adaptive Golomb rice coding are fast and effective encoder.

2.9 Compression

There are several existing image compression algorithms available. Popular and widely used compression algorithms are described in this section. The performance of the proposed algorithm is compared with them in the later section.

2.9.1 JPEG-LS: DCT-based image coding standard

JPEG-LS is widely accepted image compression algorithm. It is prediction based algorithm. JPEG-LS is simple and lower storage and computational requirement [21]. The block diagram of the JPEG-LS is shown in Figure 2-11. The prediction is computed based on the four neighbouring pixels as shown in Figure 2-11 (a, b, c and d). As neighbouring pixels in CFA are sampled from different color planes, they show varying level of pixel intensities. Therefore, direct compression of CFA image using JPEG-LS results in an inferior compression performance.

There are several solutions to this issue and few of them already discussed in Chapter 1. Structure separation and rearrangement are the most popular modifications applied in the prior compression algorithm.

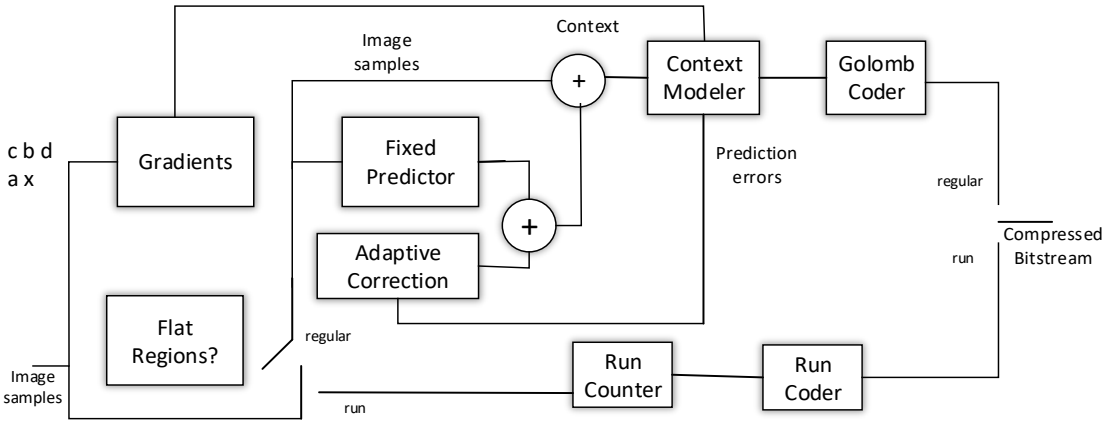


Figure 2-11: Block diagram JPEG-LS

2.9.2 JPEG-2K

The fundamental building blocks of a typical JPEG2000 encoder are shown in Figure 2-12 [22]. These components include pre-processing, DWT, quantization, arithmetic coding (tier-1 coding), and bit stream organization. The input image to JPEG2000 may contain one or more components. Although a typical color image would have three components (e.g., RGB or YCbCr), up to 16 384 (214) components can be specified for an input image to accommodate multi-spectral or other types of imagery.

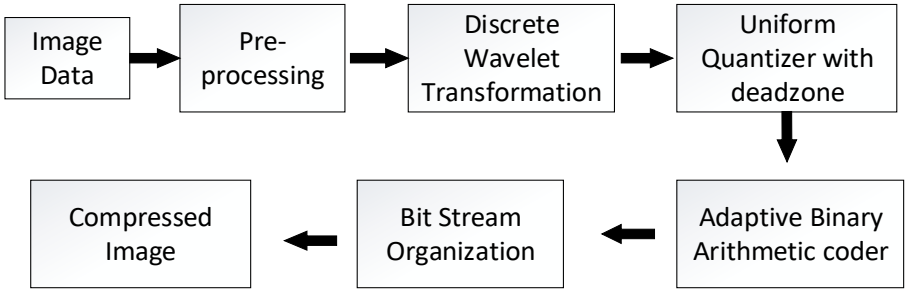


Figure 2-12: JPEG-2000 Block diagram

2.9.3 LCMI: Lossless compression of mosaic image

LCMI is focused on spectral de-correlation of spatially interleaved R, G, B samples [15]. It can remove the statistical redundancies in both spectral and spatial domains. LCMI uses wavelet

decomposition scheme, called Mallat wavelet packet transform for the de-correlation of the mosaic data. The coefficient of Mallat wavelet packet transform is compressed using adaptive Golomb rice coding technique.

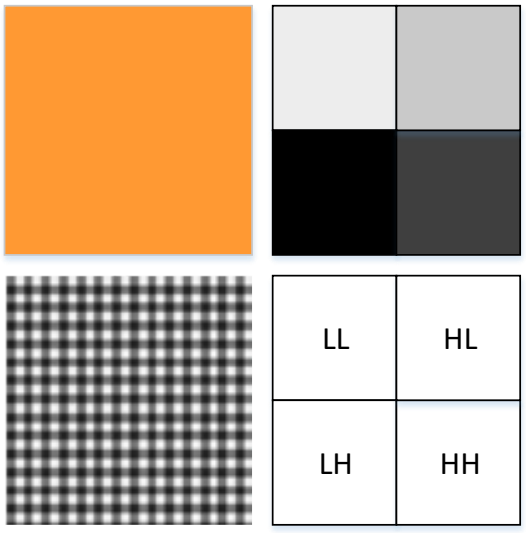
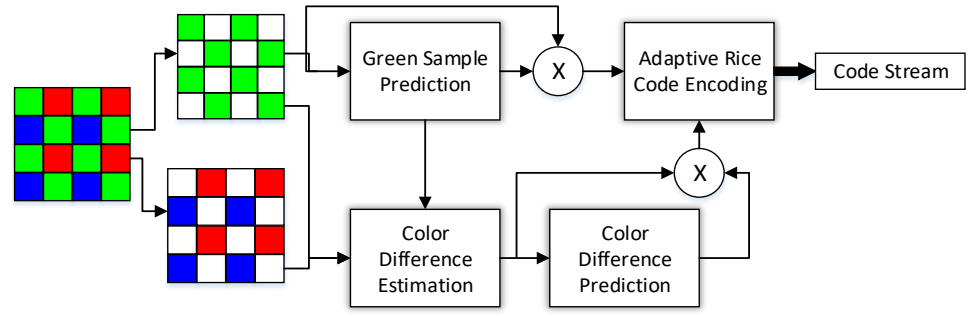


Figure 2-13: Efficiency of representing Bayer pattern mosaic image in wavelet domain

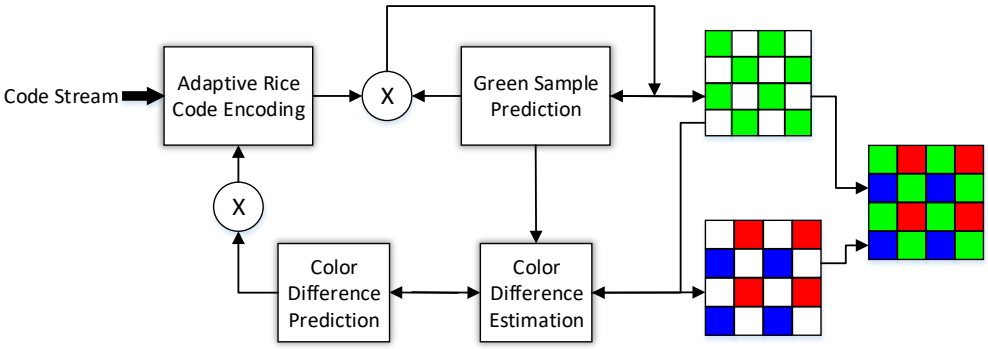
The wavelet Bayer pattern is shown in Figure 2-13. The Green channel is compressed by de-interleaved transformation which is lossless. Red and Blue channels are sampled with JPEG-LS. Fast context based coefficient coding is applied to compress the integer wavelet coefficient. The performance of LCMI is compared with JPEG-LS and JPEG-2000. This codec outperforms the JPEG-LS and JPEG-2000 in both bit rate and speed. But now several existing methods outperform the LCMI.

2.9.4 CMBP compression

CMBP compression algorithm is focused on efficient prediction of neighbor pixel [9]. It exploits a context matching technique to rank the neighboring pixels. It has an adaptive color difference



(a)



(b)

Figure 2-14: Block Diagram of CMBP compression algorithm (a) Compression (b) De-compression

estimator to remove the color spectral redundancy for the red and blue pixels. The block diagram of CMBP compression is shown in Figure 2-14. The image data is encoded by adaptive Golomb rice code. The divisor of the Rice code is adjusted with the sample residue value of the pixels. The compression algorithm outperforms the JPEG-LS, JPEG-2000 and LCMI. The complexity of the context matching is higher than the other algorithm. The proposed algorithm is compared with CMBP.

2.9.5 HP: Hierarchical prediction

Hierarchical prediction and context modeling performs best in the state of the art [3]. It consists of hierarchical predictor and context adaptive arithmetic encoder. The block diagram of hierarchical prediction model is shown in Figure 2-15. In hierarchical prediction, the Bayer CFA image is sub-sampled with two green, blue and red sub images. One green sub image is encoded by a

conventional grayscale encoder and then is used to predict the green pixels in the one. Both green sets are used to predict the reds. Then green and red pixels are used to predict the blue. The predicted values are used to find the magnitude of the prediction error. To further reduce the results bits, arithmetic adaptive encoding is applied. It performs better than the previously published works.

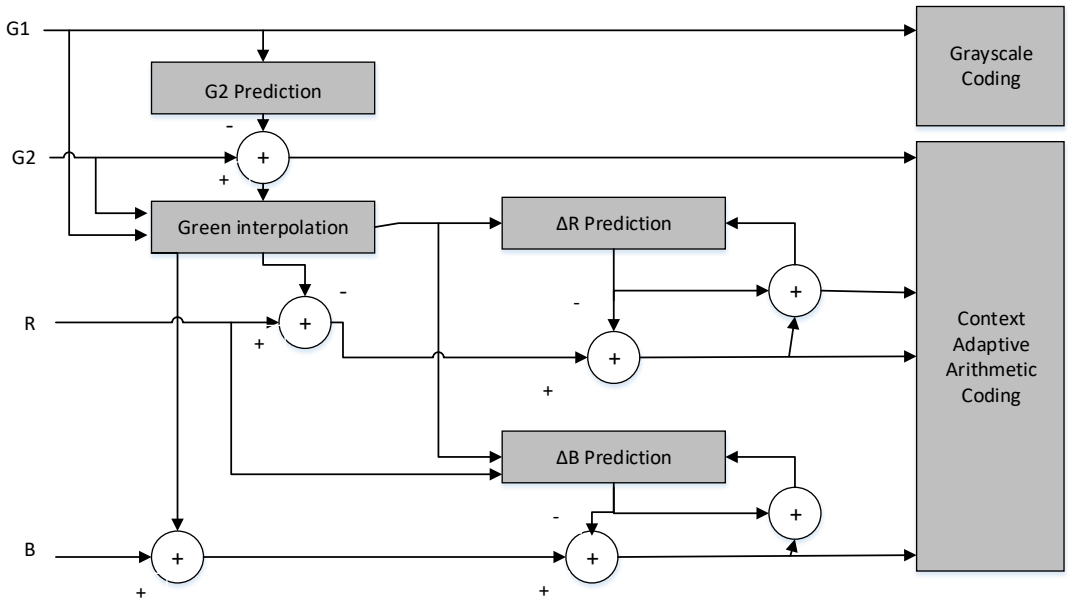


Figure 2-15: Block diagram of Hierarchical Prediction Model

Chapter 3 PROPOSED COMPRESSION DESIGN

Image compression is the key component of image processing, and in the digital world, image compression is receiving increasing attention. Several criteria must be fulfilled in order to design an efficient image compressor. Bandwidth and power are the key constraints for compression design and implementation. The proposed compression algorithms are realized with two new color transformations: $YC_MC_EC_F$ and $YC_DC_MC_O$. Proposed algorithm 1 is developed with $YC_MC_EC_F$ color transformation and proposed algorithm 2 is using $YC_DC_MC_O$ color transformation. The proposed algorithm block diagram is shown in Figure 3-1.

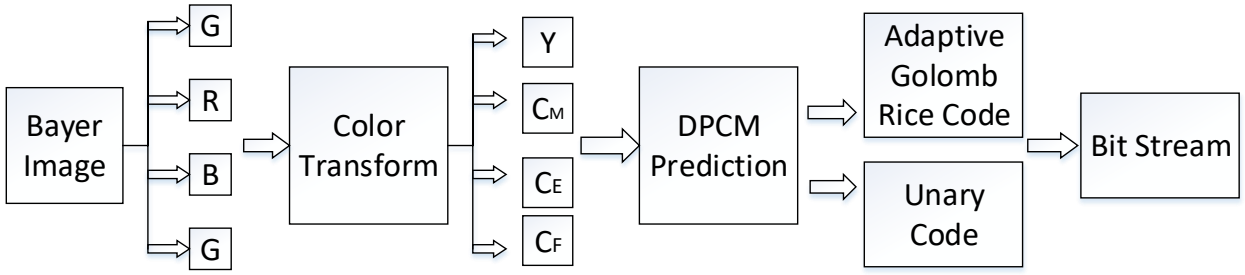


Figure 3-1: Block diagram of the proposed algorithm.

3.1 Four channel color transformation

A color transformation is performed to improve the capability of an image compression design. Well-developed color transformation provides a good image compression. CFA image compression needs color transformation to be efficient [3], [4]. Therefore, the proposed compression design introduces $YC_MC_EC_F$ and $YC_DC_MC_O$ color transformations.

The motivation for color transformation is the correlation between the Bayer RGB color components, standard deviation and entropy in each component. A significant inter color redundancy exists between the color components in CFA images. In most of the designs, the error signals generated by the individual processing of color components using a prediction model remove the intra-color redundancy. These error signals still generate significant inter-color

redundancy. The average value of the correlation for different color transformation is shown in Table 3-1, using the KODAK, D90, Olympus-EP1, and real CFA data sets. The correlation coefficient shows that significant spectral redundancy exists in Bayer RGB color transformation. The correlation result shows that the color transformation removes the spectral redundancy where YDgCoCg performs better. On the other hand, the performance of prediction based image compression depends on the variance of error signals [49]. The independent encoding of the color component depends on the smoothness. Smooth color components give only small error variance and can be coded with fewer bits. Therefore, each component's standard deviation and entropy is analyzed to obtain efficient color transformation.

Table 3-1: Average value of correlation coefficient

	G_rRBG_b	YCbCr	YDgCoCg	YLMN	$YC_MC_EC_F$	$YC_DC_MC_O$
Correlation	**0.8459**	0.4068	**0.1717**	0.3556	0.3962	0.2553

$$Y = \frac{R}{4} + \frac{Gr + Gb}{4} + \frac{B}{4} \tag{3.1}$$

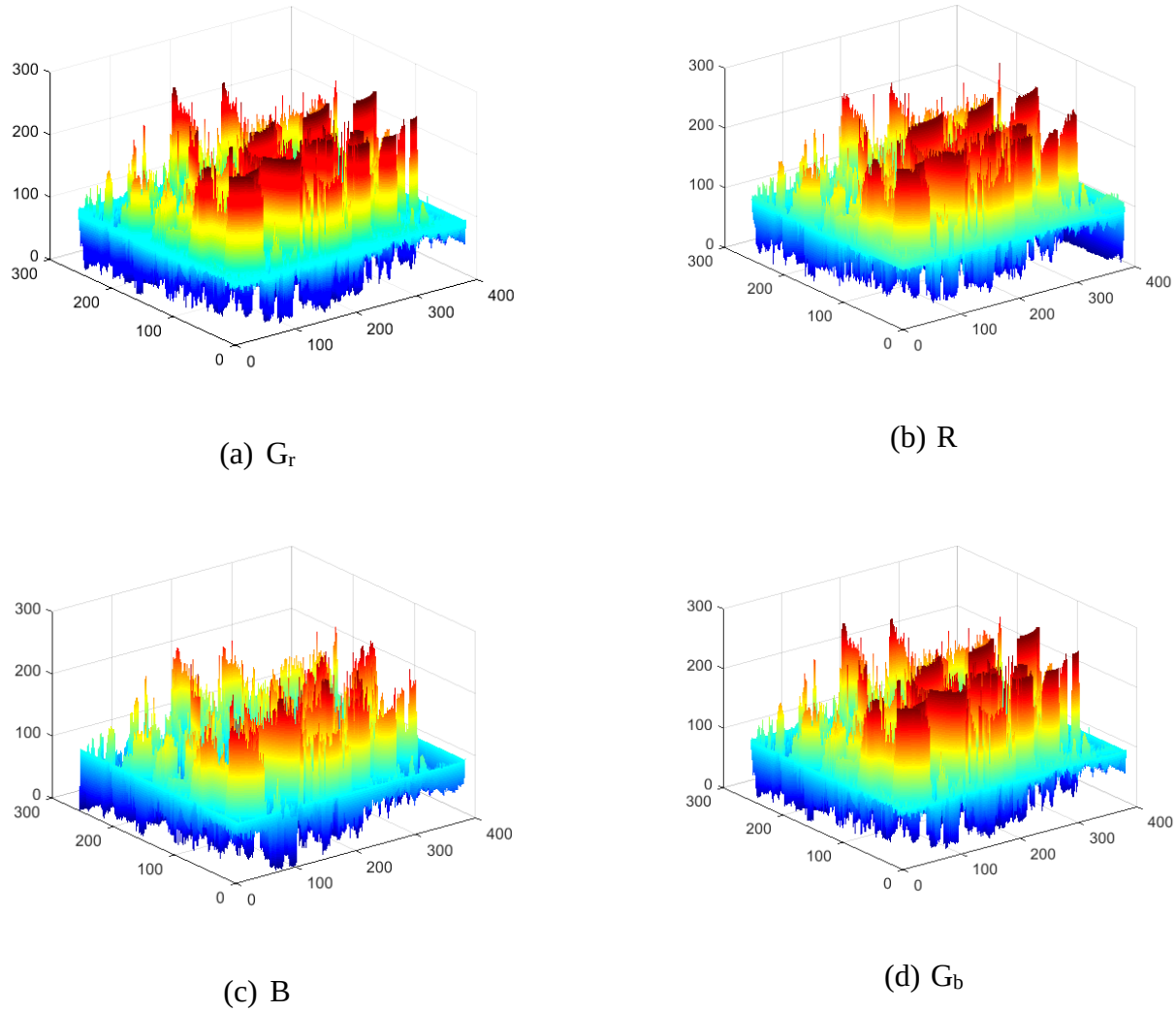
$$C_M = \frac{Gr}{8} - \frac{B}{8} \tag{3.2}$$

$$C_E = \frac{Y}{8} - \frac{Gr + Gb}{4} = \frac{R + B}{8} - \frac{Gr + Gb}{8} \tag{3.3}$$

$$C_F = \frac{Y}{8} - \frac{Gr + Gb}{16} - \frac{3B}{8} \tag{3.4}$$

Figure 3-2 shows the intensity value for each color component of Bayer RGB color space. The luminance component Y is the average of all the Bayer RGB components. The chrominance information can be generated as smoothly as possible from the difference between Bayer RGB color components. The luminance component stores mostly the information for green components;

therefore, we generate the C_E component by differentiating the average of green components from the luminance component. C_F stores the difference between luminance and blue information.



(a) G_r (b) R

(c) B (d) G_b

Figure 3-2: Dynamic range of intensity of Bayer CFA components (a-d) as G_r, R, B and G_b respectively.

The correlation between Gr and R is 0.9157, which is the maximum among the other correlation coefficients, hence, the difference between them is stored in the C_M component. The correlation between C_E and C_F is high compared to correlation between C_M and C_F or C_M and C_F. On the other hand, the C_O and C_D components store the orange chrominance information which is the difference between the information for green components. The $YC_DC_MC_O$ color transformation is better for

spectral redundancy. The following section describes in detail the evaluation of each color component.

3.1.1 $YC_MC_EC_F$ color transformation

The $YC_MC_EC_F$ color transformation is adapted from the YEF color space which is a three-channel color space for the RGB full color image [25]. The proposed color space is a four-channel color space for Bayer color images. The YEF color space is suitable for WCE (wireless capsule endoscopic) image compression [26]. In WCE images, the pixel to pixel correlation is higher. In the Bayer images, each pixel contains single color information from RGB color space. If the green pixel is recorded, then the other two will be interpolated from the neighbor's pixel value.

$$\begin{bmatrix} Y \\ C_M \\ C_E \\ C_F \end{bmatrix} = \begin{bmatrix} 1/4 & 1/4 & 1/4 & 1/4 \\ 1/8 & -1/8 & 0 & 0 \\ -1/8 & 1/8 & 1/8 & -1/8 \\ 1/16 & 1/8 & -1/4 & 1/16 \end{bmatrix} \begin{bmatrix} Gr \\ R \\ B \\ Gb \end{bmatrix} \tag{3.1}$$

$$\begin{bmatrix} Gr \\ R \\ B \\ Gb \end{bmatrix} = \begin{bmatrix} 1 & 8 & 4/3 & 8/3 \\ 1 & 0 & 4/3 & 8/3 \\ 1 & 0 & 2/3 & -8/3 \\ 1 & -8 & -22/3 & -8/3 \end{bmatrix} \begin{bmatrix} Y \\ C_M \\ C_E \\ C_F \end{bmatrix} \tag{3.2}$$

The color transformation is realized with Eqn. 3.1. The reversible matrix is shown in Eqn. 3.2, and the color components are shown in Figure 3-3. As can be seen, the color components in the Bayer images have higher correlation among Bayer RGB components. The motivation for the work is to develop a color space that will contain the lowest correlation among the color components. As a result, this proposed color space will provide the best compression for the Bayer color images.

(a) Y

(b) C_M

(c) C_E

(d) C_F

Figure 3-3: The $YC_DC_MC_O$ color components of CFA image

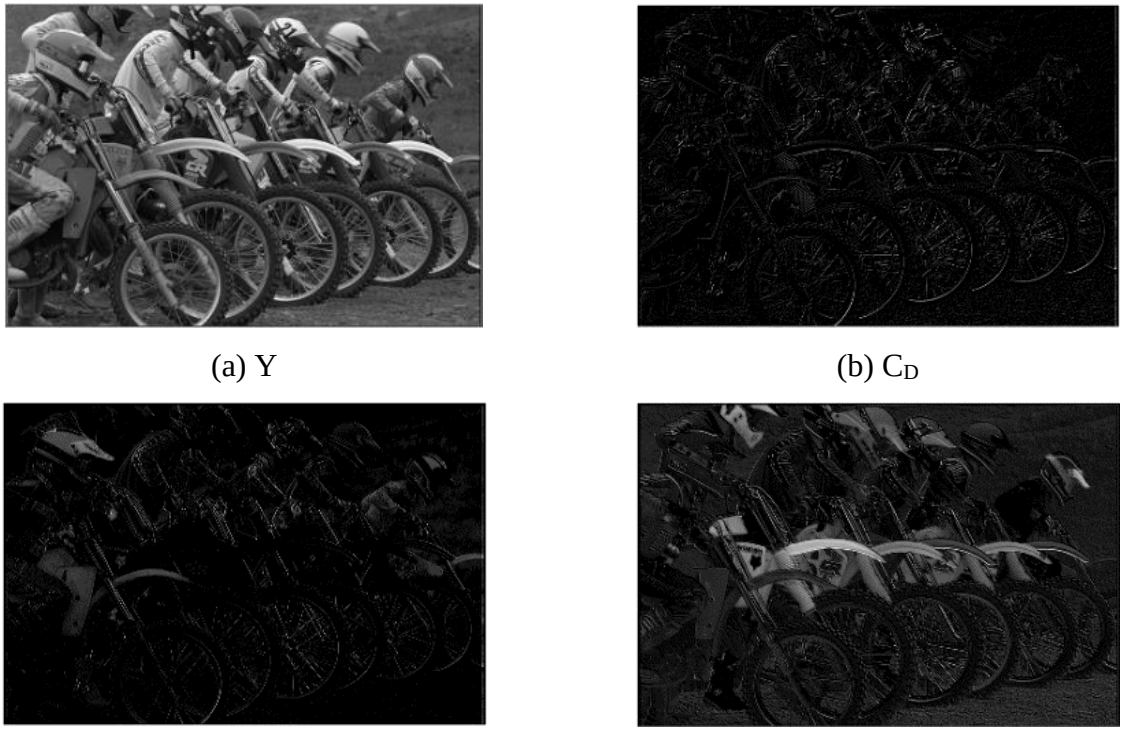
3.1.2 $YC_DC_MC_O$ color transformation

As color transformation is a key feature of the compression algorithm [4] – [13], the proposed color transformation is extended using the better correlation among the Bayer color components. A different combination of components has been used for developing a four-channel color transformation such as R and G_b, G_r and B, (G_r + B) and (G_r + R). But comparison of the entropy, standard deviation and the overall compression shows that the proposed $YC_DC_MC_O$ color transformation is efficient. The comparison is shown in Table I. The $YC_DC_MC_O$ color transformation uses Eqn. 3.3 for the transformation and the reverse matrix of the $YC_DC_MC_O$ color transformation is show in Eqn. 3.4.

$$\begin{bmatrix} Y \\ C_D \\ C_M \\ C_O \end{bmatrix} = \begin{bmatrix} 1/4 & 1/4 & 1/4 & 1/4 \\ 1/8 & 0 & 0 & -1/8 \\ 1/8 & -1/8 & 0 & 0 \\ 0 & 1/8 & -1/8 & 0 \end{bmatrix} \begin{bmatrix} Gr \\ R \\ B \\ Gb \end{bmatrix} \tag{3.3}$$

$$\begin{bmatrix} Gr \\ R \\ B \\ Gb \end{bmatrix} = \begin{bmatrix} 1 & 2 & 4 & 2 \\ 1 & 2 & -4 & 2 \\ 1 & 2 & -4 & -6 \\ 1 & -6 & 4 & 2 \end{bmatrix} \begin{bmatrix} Y \\ C_D \\ C_M \\ C_O \end{bmatrix} \quad (3.4)$$

The color components of $YC_DC_MC_O$ are shown in Figure 3-4.



(a) Y (b) C_D

(c) C_M (d) C_O

Figure 3-4: The color components of $YC_DC_MC_O$ for CFA image

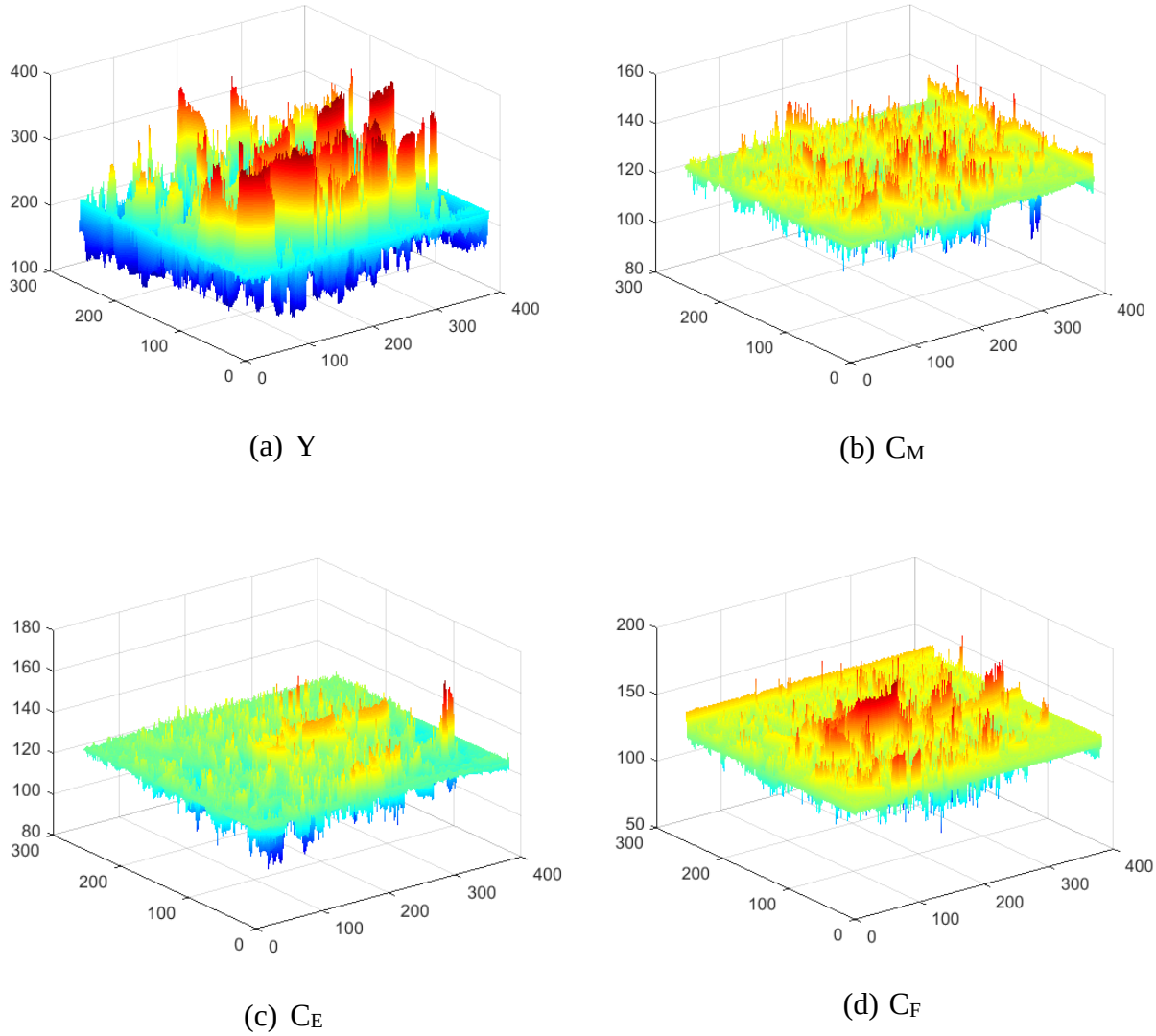
3.1.3 Feature evaluation of color components parameter

The evaluation of the above two-color transformation is measured based on each channel. The Bayer image contains the 2 x 2 macro block four-channel components known as Gr, R, B and G_b. The four components of the Bayer image are transformed into Y, C_M, C_E, C_F and Y, C_D, C_M, C_O color components. The reversible color transformation can be achieved using their reversible matrix. Each channel feature has been extracted for the comparison. The dynamic intensity

difference among the components defines the compression capabilities. Standard deviation and entropy of the components contributes to the compression design. Therefore, the dynamic intensity, correlation, standard deviation and entropy of the color components are analyzed to develop the color transformations.

3.1.3.1 Dynamic intensity of color components

The intensity of the color components is the visual assessment of the correlation among the components. The dynamic intensity range of the Bayer color image components is measured to



(a) Y (b) C_M

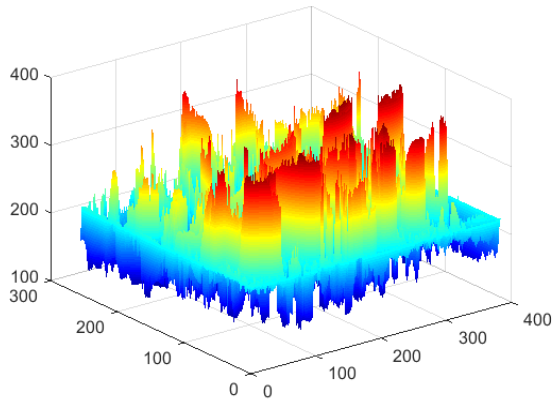
(c) C_E (d) C_F

Figure 3-5: Dynamic intensity range of proposed color transformation components of $YC_MC_EC_F$.

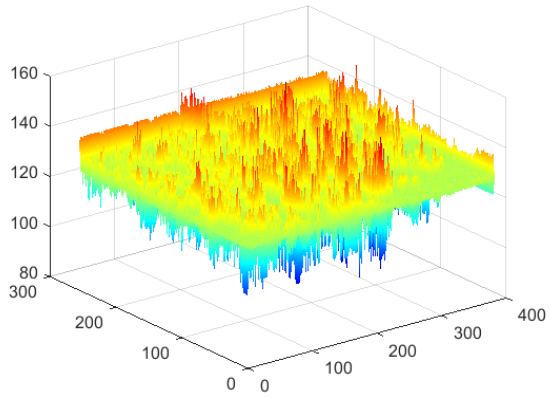
find the correlation among the components. The dynamic range of the intensity of the Bayer four-color components are shown in Figure 3-2.

Among the Bayer color components, the data samples produced by the CFA are strongly correlated. The Bayer color components need a de-correlated transformation that will give data compaction and an efficient image compression. The proposed color transformation maps a set of original {G_r, R, B, G_b} values into one approximate luminance value and three new kinds of chrominance channels: {Y, C_M, C_E, C_F} and {Y, C_D, C_M, C_O}. The de-correlation can be visually detected from their intensity plot. The intensity of the proposed color components is shown in Figure 3-5 (a-d) and Figure 3-6 (a-d) for $YC_MC_EC_F$ and $YC_DC_MC_O$ color transformation respectively.

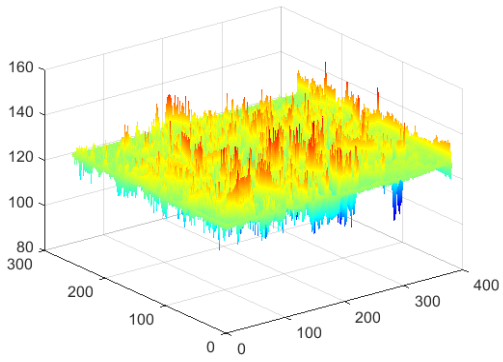
Figure 3-5 and Figure 3-6 show that, the dynamic range of intensity of the proposed color spaces is flat and the sample data are less correlated than the Bayer components. Each channel provides a dynamic range of intensity values.



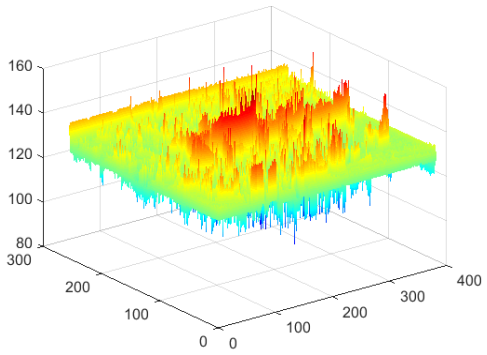
(a) Y



(b) C_D



(c) C_M



(d) C_O

Figure 3-6: Dynamic intensity of proposed color transformation components of $YC_DC_MC_O$.

The components are flat, which indicates that the compression quality of the design will be high. Maximum information is preserved in the luminance component, whereas the other three chrominance components contain less information. The motivation for the color transformation comes from the fact that the Bayer image exhibits a similar intensity distribution for this color transformation which becomes suitable for an efficient compression algorithm.

3.1.3.2 Correlation comparison

The correlation is measured to find the redundancy among the color components. The redundancy can be estimated using the following correlation Eqn. 3.5.

$$Corr(X,Y) = \frac{\sum_{i=1}^{N}(X_i - \bar{X})(Y_i - \bar{Y})}{\sqrt{\sum_{i=1}^{N}(X_i - \bar{X})^2 \sum_{i=1}^{N}(Y_i - \bar{Y})^2}} \qquad (3.5)$$

Where N, the number of image samples and X, Y color components. The average correlation value of the Bayer color components is shown in the Figure 3-7 along with the proposed methods.

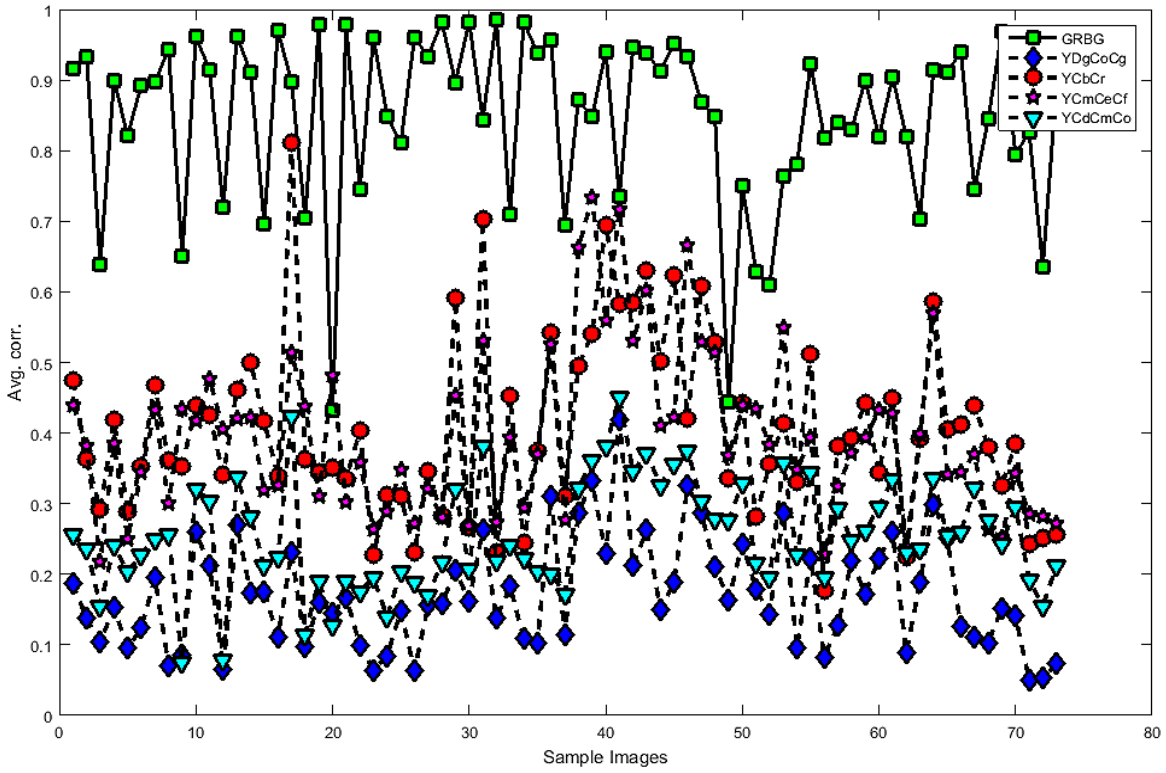


Figure 3-7: Average correlation value among the components of different color space for CFA datasets.

Bayer color components show an average correlation of 0.85 among the components where the proposed methods 1 and 2 give a correlation value of 0.3962 and 0.2553, respectively, among the components. The minimum correlation value provides the maximum compression in the design. The proposed color transformation provides a small correlation value, which will provide maximum compression.

3.1.3.3 Standard deviation comparison

The standard deviation of the proposed methods is calculated for the KODAK, Standard images, D90, real CFA and Olympus E-P1 datasets. It is a widely used parameter for the evaluation of the color transformation in terms of performance.

Table 3-2: Average standard deviation for different Bayer CFA datasets

Color-space		Standard deviation				
		kodak	D90	Standard	real_CFA	E-P1
G_rRBG_b	G	49.71	54.55	50.47	72.78	51.24
	R	48.60	54.55	44.87	72.89	57.59
	B	47.23	63.42	46.08	79.12	48.86
	G	50.14	54.56	50.09	72.54	51.24
Y_1Y_2CbCr	Y1	45.88	53.97	43.29	73.38	48.45
	Y2	46.12	53.97	43.24	73.26	48.45
	Cb	11.28	11.80	13.29	10.54	14.23
	Cr	12.67	12.02	18.25	7.63	17.61
YDgCoCg	Y	45.69	53.92	42.88	73.24	48.38
	Dg	20.17	8.50	21.80	13.19	9.46
	Co	36.50	41.77	48.18	27.27	37.82
	Cg	15.77	10.87	21.18	12.28	25.27
YLMN	Y	45.69	53.92	42.88	73.24	48.38
	L	22.14	21.56	27.49	15.05	19.69
	M	26.61	24.37	38.03	16.78	35.60
	N	23.97	23.92	26.92	22.26	28.87
$YC_MC_EC_F$	Y	**45.69**	**53.92**	**42.88**	**73.24**	**48.38**
	C_M	**3.38**	**3.10**	**4.79**	**2.14**	**4.49**
	C_E	**3.95**	**2.73**	**5.30**	**3.08**	**6.33**
	C_F	**6.90**	**7.91**	**8.70**	**5.84**	**7.39**
$YC_DC_MC_O$	Y	**45.69**	**53.92**	**42.88**	**73.24**	**48.38**
	C_D	**2.57**	**1.12**	**2.75**	**1.70**	**1.23**
	C_M	**3.38**	**1.12**	**4.79**	**2.14**	**4.49**
	C_O	**4.60**	**5.26**	**6.06**	**3.45**	**4.75**

The color transformation performance is also measured from the standard deviation and entropy of the color components. Lower value of the standard deviation and entropy gives good compression. Table 3-2 shows that the standard deviation for the proposed method provides the lowest value and the lowest value of each component is bold in the table. The Bayer color components has average standard deviation of 48.92, whereas the proposed methods 1 and 2 give

an average standard deviation of 14.98 and 14.06, respectively. The estimation of standard deviation shows that the proposed color spaces will provide good compression.

3.1.3.4 Entropy comparison

Table 3-3: Comparison of entropy of different color transformation of CFA images

Color-space		Entropy				
		kodak	D90	Standard	real_CFA	E-P1
G_rRBG_b	G_r	7.11	7.41	6.74	6.74	7.11
	R	7.11	7.42	6.74	6.79	7.36
	B	7.11	7.24	6.62	6.27	6.9
	G_b	7.12	7.41	6.73	6.74	7.11
Y_1Y_2CbCr	Y_1	7.12	7.38	6.94	6.37	6.69
	Y_2	7.13	7.38	6.93	6.37	6.69
	C_b	1.26	1.45	2.99	4.11	3.1
	C_r	3.38	3.82	4	3.58	5.38
$YD_gC_oC_g$	Y	7.11	7.38	6.92	6.9	7.1
	D_g	2.95	2.38	3.1	2.52	2.53
	C_o	5.48	5.34	5.01	2.07	6.26
	C_g	3.93	3.32	2.45	0.33	3.11
YLMN	Y	7.11	7.38	6.92	6.9	7.1
	L	4.81	4.71	4.45	2.06	5.42
	M	3.93	4.48	4.64	4.52	5.17
	N	5.15	4.78	3.54	0.62	4.78
$YC_MC_EC_F$	Y	**7.11**	**7.38**	**6.92**	**6.9**	**7.1**
	C_M	**1.54**	**1.24**	**1.33**	**0.56**	**0.93**
	C_E	**1.42**	**1.35**	**2.8**	**3.23**	**2.19**
	C_F	**3.78**	**3.67**	**3.24**	**0.98**	**4.01**
$YC_DC_MC_O$	Y	**7.11**	**7.38**	**6.92**	**6.9**	**7.1**
	C_D	**1.48**	**0.88**	**1.48**	**1.19**	**0.93**
	C_M	**1.54**	**1.24**	**1.33**	**0.56**	**0.93**
	C_O	**3.17**	**3.17**	**2.99**	**1.19**	**3.55**

The luminance component Y has the similar entropy value of Bayer RGB components which indicates that the compression of this component will be like the other Bayer color components. The entropy values of the proposed $YC_MC_EC_F$ and $YC_DC_MC_O$ color components are tabulated in Table 3-3 and the lowest value of each component is bold in the table.

For the three chrominance components, the entropy value is less than it is for the other color components; hence, it takes less bits per pixel compares to the other components. The proposed methods outperform the other color components. The performance parameters of color transformation give good results for the proposed work in the state of the arts. The dynamic intensity, correlation value, entropy and the standard deviation show that proposed color transformations outperform the existing color transformation for compression algorithms.

3.1.4 Prediction gain comparison

The compression performance depends on the prediction model, as the accuracy of the prediction should be high. In previous work, several complex prediction models were used, such as template matching, hierarchical prediction [3], and context matching [6] in lossless CFA image compression. The disadvantages of these models are a higher memory requirement and computational complexity. Therefore, it is convenient to use a simple prediction model to increase the coding performance of the compression algorithm.

The simple delta pulse coded modulation (DPCM) prediction model shows better performance in terms of computational complexity and lossless compression design in smooth images [19]. The comparison of the prediction model with different color transformation is shown in Figure 3-8. The proposed two-color transformations give the highest prediction gain in comparison with other widely used color transformations. This result signifies that the proposed color transformations will be efficient for the compression design with a DPCM prediction model.

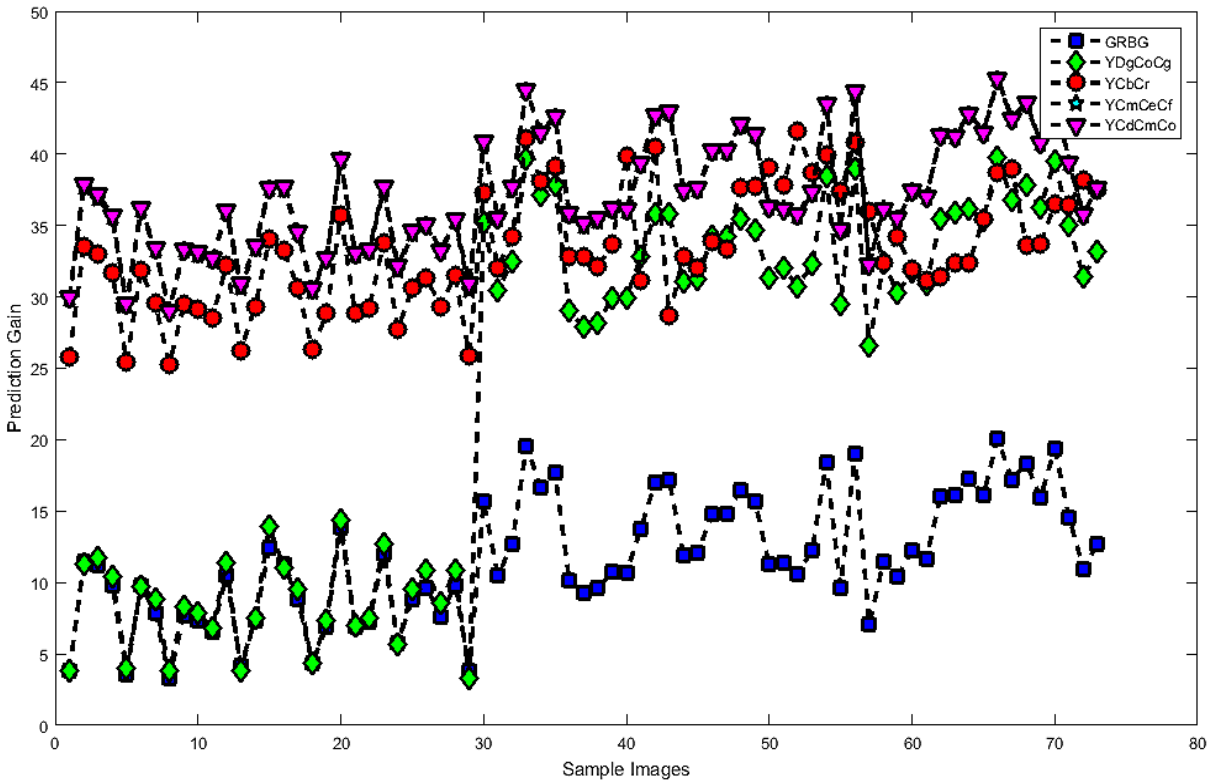


Figure 3-8: Prediction gain comparison with different color spaces

3.1.5 Lossless color transformation

Color transformation needs to be fully reversible for lossless image compression [27]. The proposed color transformations can be made fully reversible by adding extra bits in the fraction parts along with integer 8 bits. Table 3-4 shows that each extra bit in the fraction parts gives better PSNR (Peak Signal-to-Noise Ratio) and SSIM (Structural Similarity) index. Three extra bits in the fractional part makes the color transformation fully reversible.

The hardware implementation will cost an extra three bits for making the proposed color transformation lossless. In terms of the compression performance, the proposed color transformations are developed as lossless in this work. Therefore, the proposed compression algorithm becomes a lossless compression design.

Table 3-4: Average image quality index with different bit length for KODAK Dataset

Number of bits + fractional part	$YC_MC_EC_F$		$YC_DC_MC_O$	
	Overall PSNR	SSIM	Overall PSNR	SSIM
8+0	41.15	0.9737	42.92	0.9819
8+1	46.96	0.9921	47.96	0.9943
8+2	52.37	0.9977	52.82	0.9980
8+3	Inf	1.0000	Inf	1.0000

3.2 Structure separation

Structure separation is necessary for the Bayer color filter array images. The prediction is made from the neighbouring pixels, but the Bayer pattern does not contain the same color information at the neighbouring pixel. To obtain the same color components for the prediction, structure separation is necessary.

3.2.1 $YC_MC_EC_F$ component structure separation

The input CFA image contains four color components {G_r, R, B, G_b} in each 2 x 2 macro block of CFA data. The structural separation is a necessary step for the compression algorithm because the compression-first of the CFA image is not efficient [13]. It has high artificial frequencies for the mosaic arrangement of the Bayer color pixels.

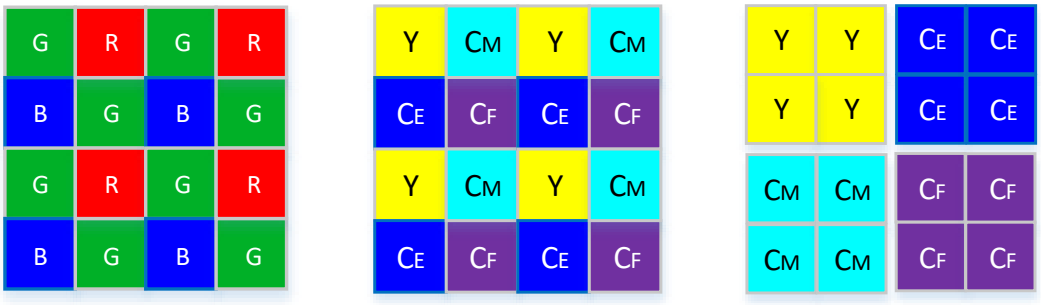


Figure 3-9: The structure separation from Bayer color transformations to the proposed $YC_MC_EC_F$ color transformation.

As the CFA image is down sampled into four separate color channels, the proposed DPCM module was applied to each color component separately and, the structure separation was achieved by shifting G to the left and R and B to the right of the frame. The structural separation is shown in Figure 3-9 for the $YC_MC_EC_F$ color transformation. The structural separation will give smooth data to be sampled by the DPCM module.

3.2.2 $YC_DC_MC_O$ component structure separation

The $YC_DC_MC_O$ color transformation is also processed with structure separation. Figure 3-10 shows the different components of $YC_DC_MC_O$ color transform after structure separation.

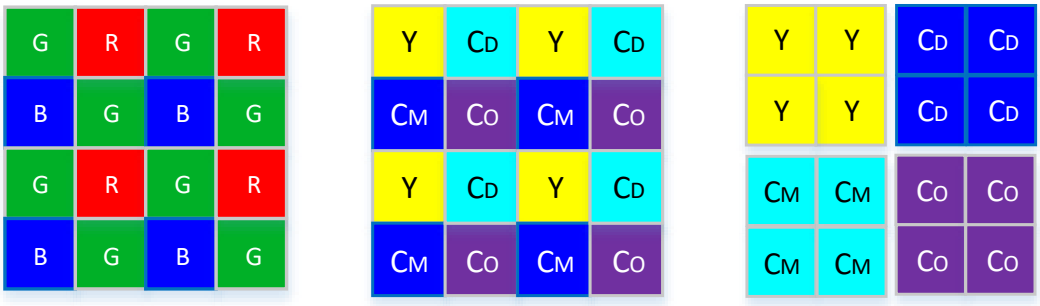


Figure 3-10: The structure separation from Bayer color transformation to $YC_DC_MC_O$ color transformation

3.3 Prediction model

In the proposed compression design, the DPCM module was applied in a different order. Eqn. 3.6 makes the prediction for all the columns except for the first column. To minimize the prediction error for the first column, Eqn. 3.7 is used to predict the error. Then the DPCM module sends the data to the encoder module.

$$dX(i,j) = X(i,j) - X(i,j-1) \qquad (3.6)$$

$$dX(1,i) = X(1,j) - X(1,j-1) \qquad (3.7)$$

The proposed color transformation provides de-interleaved sub images, which encode independently and give the optimal compression for the CFA images.

$$M \leftarrow 2 * dX \tag{3.8}$$

$$M \leftarrow 2 * abs(dX) - 1 \tag{3.9}$$

The prediction model is complex in HP [3], CMBP [9] and LCMI [15]. The proposed algorithm shows a better result with a simple modified prediction model. Therefore, the complexity is compared with the existing simple prediction models such as median edge prediction model 1 and 2, YLMN [13], and so on.

Table 3-5: Number of operations per pixel required for various prediction model

Prediction Model	ADD	SHIFT	CMP	Total	MEM
MEP1	2	0	7	9	2 x Width
MEP2	3	1	4	8	2 x Width
DPCM	1	0	0	1	0
YLMN	2.5	0	0	2.5	½ x Width
$YC_MC_EC_F$	2.5	1.5	0	4.0	1 x Width
$YC_DC_MC_O$	1.5	1	0	2.5	1 x Width

The computational complexity is compared in Table 3-5 for the prediction model of the color transformations of $YC_MC_EC_F$ and $YC_DC_MC_O$. The proposed method offers the best result in comparison to the other methods, except for YLMN. DPCM hardware for color transformation requires 4 and 2.5 operation per pixel for $YC_MC_EC_F$ and $YC_DC_MC_O$ respectively which is acceptable in consideration of complexity and compression results.

3.4 Encoding

The adaptive Golomb rice encoding method is used in the proposed compression algorithm. Other encoding methods such as Huffman coding, LZW requires extra memory. In [50], LZW needs to build content addressable memory and store the coder dictionary [51]. In [52], Huffman coding is used for compression algorithm. Huffman table requires memory to implement the Huffman coding [53]. A DPCM and Wavelet based compression algorithm is discussed in [54]. It requires pixels in blocks which causes memory and Wavelet transform is computationally expensive. Proposed compression algorithm is built to minimize the complexity and lower the implementation cost. Considering the low complexity and low power design, Unary coding, Golomb Rice or adaptive Golomb Rice does not require any memory. Therefore, the performance of Unary encoding, Golomb rice coding and adaptive Golomb Rice coding has been evaluated with sample KODAK images, to minimize the design cost.

3.4.1 Unary coding

Table 3-6: Bit rate of proposed compression algorithm using Unary coding for each color component of $YC_DC_MC_O$ color transformation in BPP.

Kodak	Y	C_D	C_M	C_O	Avg.
1	16.31	5.80	5.75	6.31	8.54
2	8.54	2.67	2.98	3.37	4.39
3	7.39	2.17	2.41	2.80	3.69
4	10.36	3.17	3.16	3.22	4.98
5	17.73	6.04	6.04	6.56	9.09
6	11.78	4.33	3.61	4.22	5.98
7	9.21	2.77	2.70	2.85	4.38
8	17.75	8.17	7.51	7.80	10.31
9	9.04	3.12	2.98	3.15	4.57
10	9.74	3.11	2.98	3.20	4.76
Avg.	11.79	4.13	4.01	4.35	6.07

Unary coding is considered as representative of the natural number [46]. It represents the value as the number of 1s followed by a zero. The performance of the unary coding is shown in Table 3-6. Unary coding is special in the case of Golomb rice coding.

Therefore, the proposed algorithm is also realized with Golomb rice coding which performs better than Unary coding. The value of k in the Golomb rice coding is set to 0 to obtain the Unary coding from the Golomb rice coding.

3.4.2 Golomb Rice coding

Golomb rice coding is a lossless data compression encoding method. The algorithm needs a constant of the power of 2 such as (2^k) [48]. The Golomb rice code can be measured using Eqn. 3.10 and 3.11.

$$quotient\_code \leftarrow \frac{M}{2^k} \tag{3.10}$$

$$code\_stream \leftarrow quotient\_code + k + 1 \tag{3.11}$$

The pseudo code for the Golomb rice codes is described below.

Golomb rice pseudo code:

1. Select the value of k i.e. k = 2.
2. For M, the number to be encoded
    a. Quotient = q = int[M/2^k]
    b. Remainder = r = M modulo 2^k
3. Generate Code
    a. Code format: <Quotient Code> <Remainder Code>
    b. Quotient coded as unary code (q length string of 1 bits) and a 0 bit
    c. k bits are needed for remainder.

The proposed compression algorithm is realized with Golomb rice coding for $YC_DC_MC_O$ color transformation. The constant value of k is selected for 2. The compression bit rate is shown in Table 3-7. The compression bit rate is calculated for 10 sample CFA images. A further analysis of the encoding method is exercised to obtain the optimum encoding. Therefore, the adaptive Golomb rice is also evaluated in the proposed compression algorithm. Adaptive Golomb rice coding out-performs the unary and Golomb rice coding for the proposed compression algorithm.

Table 3-7: proposed compression algorithm using Golomb Rice Coding for each color component of $YC_DC_MC_O$ color transformation in BPP

KODAK	Y	C_D	C_M	C_O	Avg.
1	8.38	3.92	3.91	4.03	5.06
2	4.87	3.22	3.27	3.35	3.68
3	4.78	3.15	3.18	3.27	3.59
4	5.60	3.32	3.31	3.32	3.89
5	9.73	3.98	3.98	4.12	5.45
6	6.18	3.59	3.42	3.55	4.18
7	5.67	3.28	3.25	3.28	3.87
8	10.79	4.59	4.41	4.45	6.06
9	5.47	3.32	3.28	3.32	3.85
10	5.71	3.31	3.28	3.34	3.91
Avg.	6.72	3.57	3.53	3.6	4.35

3.4.3 Adaptive Golomb Rice coding

The encoder is based on the adaptive Golomb rice coding and unary coding. The energy of the residual signal is reduced by the proposed color transformation as well as by the DPCM prediction model which consider both the intra-spectral and inter-spectral redundancy. The JPEG_LS algorithm used adaptive Golomb rice coding [49], which is the same as the proposed encoding method. The proposed color components contain different characteristics and statistical properties. Therefore, each component is encoded separately using a single context.

A DPCM prediction model provides the residual signal which is mapped using the Eqn. 3.8 and 3.9, respectively, for the positive and negative values. The mapping is necessary for the adaptive Golomb rice coding and unary coding which work for positive numbers only. Eqn. 3.8 and Eqn. 3.9 provide positive value to the encoder input. Then the mapped residual is represented with a unary coding of the quotient, with Eqn. 3.10 as the prefix and with Eqn. 3.11 as the fixed length coding of the remainder. The parameter k is continuously updated based on the statistics of the signal by counting the number of occurrences (N_c) and accumulated errors (A_c). An optimum threshold value ($N_{Threshold}$= 8) is used in the algorithm, which limits to the N_c and A_c by half of their current value. The resulted encoded bit stream is sent to the transmission line.
The parameter k is updated with the following pseudo code.

```
k ← 0
While Nc x 2^k ≥ Ac
        k ← k + 1
Update parameter Nc and Ac
Nc ← Nc + 1
Ac ← Ac + abs(M)
If Nc > Nthreshold
        Nc ← Nc/2
        Ac ← Ac/2
```

3.5 Compression result

The proposed algorithm exercises the adaptive Golomb rice coding on both the $YC_MC_EC_F$ and $YC_DC_MC_O$ color transformations for KODAK, D90, real CFA, Olympus-EP1. The compression results for KODAK, D90, real CFA, and Olympus-EP1 datasets are shown in Table 3-11, Table 3-8, Table 3-9, and Table 3-10 respectively.

Table 3-8: Bit rate of proposed compression algorithm for $YC_MC_EC_F$ and $YC_DC_MC_O$ color transformations for Kodak dataset in BPP.

KODADK	Y	C_M	C_E	C_F	Avg.	Y	C_D	C_M	C_O	Avg.
1	6.53	4.27	4.03	5.07	4.97	6.53	4.30	4.27	4.43	4.88
2	5.03	2.81	2.79	3.73	3.59	5.03	2.53	2.81	3.09	3.36
3	4.55	2.27	2.40	3.04	3.06	4.55	2.08	2.27	2.48	2.84
4	5.30	2.81	2.90	3.53	3.63	5.30	2.82	2.81	2.88	3.45
5	6.89	4.26	4.22	5.14	5.13	6.89	4.29	4.26	4.44	4.97
6	5.32	3.00	3.33	4.01	3.91	5.32	3.32	3.00	3.35	3.74
7	5.14	2.46	2.49	3.20	3.32	5.14	2.45	2.46	2.57	3.16
8	7.00	4.50	4.06	5.28	5.21	7.00	4.71	4.50	4.62	5.21
9	4.88	2.62	2.59	3.39	3.37	4.88	2.68	2.62	2.72	3.22
10	5.20	2.66	2.60	3.53	3.50	5.20	2.75	2.66	2.80	3.35
11	5.45	3.11	3.11	4.03	3.92	5.45	3.24	3.11	3.27	3.77
12	4.65	2.38	2.51	3.24	3.19	4.65	2.39	2.38	2.56	2.99
13	6.78	4.52	4.75	5.34	5.35	6.78	4.70	4.52	4.67	5.17
14	6.12	3.45	3.63	4.52	4.43	6.12	3.67	3.45	3.78	4.25
15	5.16	2.87	2.92	3.53	3.62	5.16	2.75	2.87	2.93	3.43
16	4.71	2.42	2.68	3.36	3.29	4.71	2.62	2.42	2.71	3.12
17	5.49	2.74	2.73	3.55	3.63	5.49	2.78	2.74	2.80	3.45
18	6.14	3.71	3.84	4.53	4.55	6.14	3.81	3.71	3.78	4.36
19	5.69	3.18	3.14	3.91	3.98	5.69	3.26	3.18	3.24	3.84
20	4.20	2.36	2.62	3.33	3.13	4.20	2.42	2.36	2.66	2.91
21	5.09	2.95	3.16	3.78	3.74	5.09	3.08	2.95	3.11	3.56
22	5.76	3.34	3.32	4.19	4.15	5.76	3.35	3.34	3.41	3.96
23	4.69	2.27	2.48	3.06	3.13	4.69	2.08	2.27	2.42	2.86
24	6.31	3.72	3.50	4.51	4.51	6.31	3.84	3.72	3.77	4.41
Avg.	5.50	3.11	3.16	3.95	**3.93**	5.50	3.16	3.11	3.27	**3.76**

Table 3-9: Bit rate of proposed compression algorithm for $YC_MC_EC_F$ and $YC_DC_MC_O$ color transformations for Olympus-EP1 dataset in BPP.

E-P1	Y	C_M	C_E	C_F	Avg.	Y	C_D	C_M	C_O	Avg.
1	6.19	2.87	2.77	3.73	3.89	6.19	2.97	2.87	2.95	3.75
2	5.91	2.50	2.14	3.21	3.44	5.91	2.55	2.50	2.62	3.40
3	6.19	2.88	3.17	3.75	4.00	6.19	2.79	2.88	2.86	3.68
4	5.85	2.66	2.99	3.43	3.73	5.85	2.49	2.66	2.58	3.40
5	4.75	1.96	2.12	2.37	2.80	4.75	1.61	1.96	1.97	2.57
6	6.06	2.71	2.61	3.50	3.72	6.06	2.98	2.71	2.68	3.61
7	4.50	1.77	2.10	2.23	2.65	4.50	1.56	1.77	1.69	2.38
8	6.78	3.70	3.55	4.50	4.63	6.78	3.90	3.70	3.74	4.53
9	5.84	2.25	2.32	3.08	3.37	5.84	2.07	2.25	2.38	3.14
10	5.87	2.66	2.29	3.41	3.56	5.87	2.85	2.66	2.70	3.52
11	4.47	2.14	1.94	2.70	2.81	4.47	2.19	2.14	2.20	2.75
12	3.40	1.75	1.60	2.13	2.22	3.40	1.61	1.75	1.78	2.13
13	5.42	2.47	2.40	3.44	3.43	5.42	2.65	2.47	2.61	3.29
14	5.35	2.44	2.22	3.34	3.34	5.35	2.51	2.44	2.59	3.22
15	5.16	2.35	2.15	3.21	3.22	5.16	2.44	2.35	2.42	3.09
16	5.40	2.33	2.30	3.36	3.35	5.40	2.46	2.33	2.49	3.17
17	3.92	1.76	1.81	2.11	2.40	3.92	1.42	1.76	1.81	2.23
18	4.23	2.00	2.03	2.33	2.65	4.23	1.59	2.00	2.03	2.46
19	3.31	1.52	1.51	1.94	2.07	3.31	1.32	1.52	1.60	1.94
20	3.70	1.68	1.60	2.09	2.27	3.70	1.48	1.68	1.74	2.15
21	2.98	1.49	1.46	1.74	1.92	2.98	1.20	1.49	1.48	1.79
22	3.56	1.69	1.59	2.07	2.23	3.56	1.42	1.69	1.66	2.08
23	4.82	2.20	2.07	2.92	3.00	4.82	2.04	2.20	2.40	2.87
24	4.09	1.85	1.85	2.29	2.52	4.09	1.64	1.85	1.91	2.37
Avg.	4.91	2.23	2.19	2.87	3.05	4.91	2.16	2.23	2.29	2.90

Table 3-10: Bit rate of proposed compression algorithm for $YC_MC_EC_F$ and $YC_DC_MC_O$ color transformations for real CFA dataset in BPP.

Real CFA	Y	C_M	C_E	C_F	Avg.	Y	C_D	C_M	C_O	Avg.
1	5.81	3.06	3.41	4.23	4.13	5.81	3.17	3.06	3.49	3.88
2	5.80	3.18	3.51	4.25	4.18	5.80	3.30	3.18	3.55	3.96
3	5.69	3.12	3.44	4.18	4.11	5.69	3.19	3.12	3.51	3.88
4	4.94	2.34	2.58	3.38	3.31	4.94	2.34	2.34	2.60	3.05
5	4.80	2.25	2.52	3.28	3.21	4.80	2.26	2.25	2.52	2.96
6	4.95	2.61	2.87	3.16	3.40	4.95	2.76	2.61	2.44	3.19
7	4.04	1.94	2.20	2.82	2.75	4.04	1.93	1.94	2.15	2.52
8	3.96	1.90	2.15	2.77	2.69	3.96	1.90	1.90	2.10	2.47
9	5.29	2.80	3.03	3.76	3.72	5.29	2.83	2.80	3.08	3.50
10	5.28	2.79	3.03	3.75	3.71	5.28	2.83	2.79	3.08	3.49
11	4.33	2.05	2.31	2.95	2.91	4.33	2.09	2.05	2.32	2.70
12	4.30	2.03	2.30	2.93	2.89	4.30	2.07	2.03	2.31	2.68
13	4.28	2.03	2.26	2.95	2.88	4.28	2.04	2.03	2.30	2.66
14	4.56	2.24	2.50	3.23	3.13	4.56	2.28	2.24	2.56	2.91
Avg.	4.86	2.45	2.72	3.40	**3.36**	4.86	2.50	2.45	2.72	**3.13**

Table 3-11: Bit rate of proposed compression algorithm for $YC_MC_EC_F$ and $YC_DC_MC_O$ color transformations for D90 dataset in BPP.

D90	Y	C_M	C_E	C_F	Avg.	Y	C_D	C_M	C_O	Avg.
1	4.65	1.78	1.77	2.65	2.71	4.65	1.60	1.78	2.03	2.51
2	6.38	3.02	2.91	4.08	4.10	6.38	3.21	3.02	3.30	3.98
3	4.58	2.11	1.98	2.88	2.89	4.58	1.98	2.11	2.37	2.76
4	3.28	1.50	1.58	1.69	2.01	3.28	1.62	1.50	1.38	1.95
5	4.48	1.95	1.83	2.43	2.67	4.48	2.04	1.95	1.97	2.61
6	3.09	1.46	1.48	2.04	2.02	3.09	1.33	1.46	1.64	1.88
Avg.	4.41	1.97	1.92	2.63	**2.73**	4.41	1.96	1.97	2.12	**2.62**

3.6 Summary

The proposed color transformations show efficient features for the compression algorithm. The compression bit rate is measured at 3.05 bpp and 2.90 bpp for the Olympus-EP1 dataset for $YC_MC_EC_F$ and $YC_DC_MC_O$ compression algorithms, respectively. The bit rate for kodak data set is 3.93 bpp and 3.76 bpp, for real CFA 3.36 bpp and 3.13 bpp, and for D90 2.73 bpp and 2.67 bpp for the $YC_MC_EC_F$ and $YC_DC_MC_O$ compression algorithms, respectively. The proposed compression algorithm is also extended for the WCE image features [28] – [39]. The proposed color transformations are developed considering the prediction variance. Therefore, a smooth image will perform better with the proposed color transformation. The inter pixel correlation of a color component is high in the WCE image; therefore, the proposed color transformation gives a high compression ratio. The DPCM module is simple, efficient, and has a low implementation cost. The encoding run length or arithmetic coding is costly to implement; on the other hand, the adaptive Golomb coding is efficient and the cost is low. The proposed compression algorithm is developed using all the above features for an efficient compression algorithm for a natural image and WCE image. The performance of the proposed compression is evaluated in chapter 4.

Chapter 4 PERFORMANCE EVALUATION

4.1 Introduction

An image compression technique introduces noise into the reconstructed image [47]; therefore, it is important to evaluate the quality of the image. The evaluation can be processed through the analysis of the statistical properties of the image [40]. This section describes these statistical properties for evaluating the compression algorithm. The objective of the evaluation of the image for the compression algorithm is to measure the peak signal to noise ratio (PSNR), compression ratio (CR), and structural similarity (SSIM) index. The tradeoffs between the quality and the compression ratio depend on the application. In medical applications, the quality of the image needs to be high. The proposed algorithm works as near lossless and lossless. A lossless compression provides PSNR value as infinity and SSIM index as 1.

4.2 Evaluation parameters

The evaluation parameters are measured to compare the compression algorithm. The widely used parameters are peak signal to noise ratio, the compression ratio, and the structural similarity index. The proposed algorithm is near lossless and lossless. The evaluation parameters are measured for the near lossless compression.

4.2.1 Peak signal to noise ratio

Peak signal to noise ratio (PSNR) is the most popular parameter for the quality measurement of the compressed image. The mathematical expression is shown in Eqn. 4.1.

$$PSNR = 10\log_{10}\frac{I^2}{MSE} \tag{4.1}$$

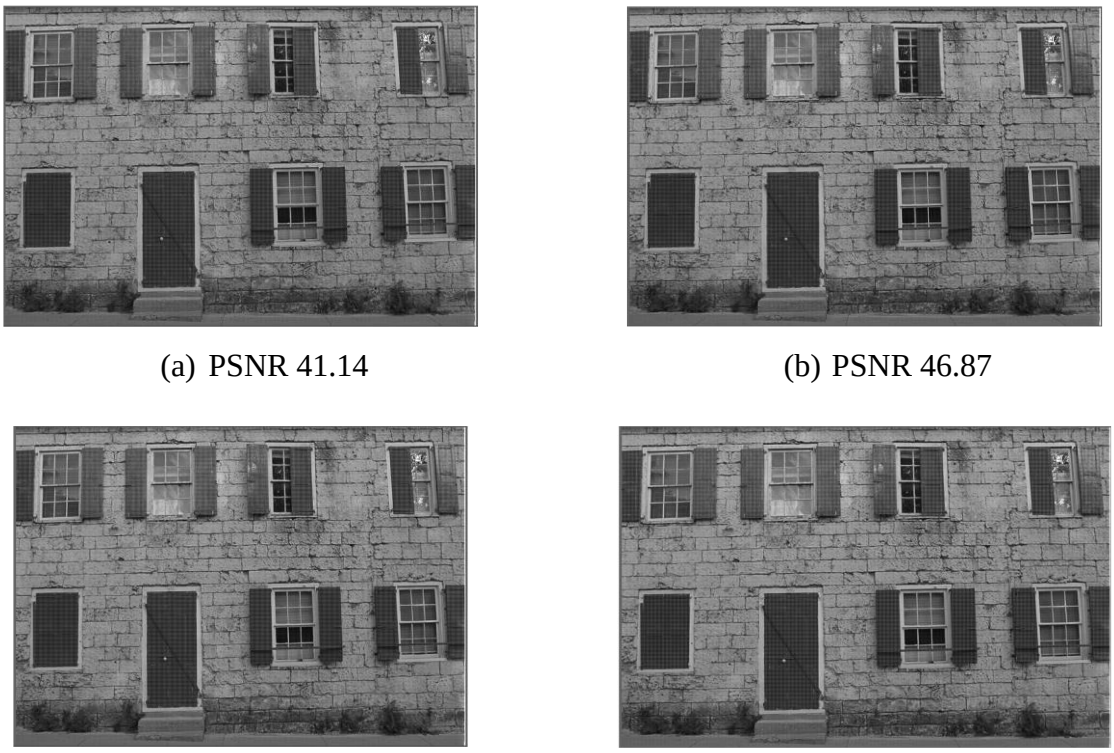
Where “I” is the intensity level of image pixels, the value of I is in the range of 0 to 255, and MSE is the Mean Square Error. The mathematical expression for MSE is in Eqn. 4.2.

$$MSE = \frac{1}{MN}\sum_{i=1}^{M}\sum_{j=1}^{N}\left(A_{i,j} - B_{i,j}\right) \tag{4.2}$$

Where, A = Original image of size M × N,

B = Reconstructed image of size M × N.

Figure 4-1 shows the PSNR comparison of near lossless image compression for the proposed algorithm $YC_MC_EC_F$.



(a) PSNR 41.14 (b) PSNR 46.87

(c) PSNR 52.14 (d) PSNR Inf

Figure 4-1 Reconstructed image PSNR value with number of extra bit in hardware 0, 1, 2, 3 bits in Figure a, b, c, and d respectively, for the proposed algorithm $YC_MC_EC_F$

4.2.2 Compression ratio

The image compression technique is based on discarding unnecessary and regenerative image information from the original image. The measurement of discarded image information is necessary to make a good compression algorithm. The performance measurement is made through the compression ratio for a design. The compression ratio is a measurement of the reduction of the detail coefficient of the data. It can be described as Eqn. 4.3.

$$CR = \frac{Discarded\ \ data}{Original\ \ data} \tag{4.3}$$

In lossless compression, CR is fixed for a certain image. Although in lossy compression, the CR can be varied to obtain a different quality of image. A lossy compression design can discard as much as details from the original image to obtain a higher compression ratio. The trade off for the high compression ratio is the poor quality of the reconstructed image. In DCT or DWT, the quantization table and scaling factor are the controlling parameters of the compression ratio. The compression design performance depends on the compression ratio. For lossless compression, it is expected to be as high as possible.

4.2.3 Structural similarity index

The SSIM is based on the assumption that the image signals are highly structured. The neighboring pixels are highly correlated. The correlation signifies information about the structure of the image in the visual perception [41]. Therefore, the image quality can be achieved from the SSIM measurement. The compression can distort the image structure. The reconstructed image can be analyzed with the SSIM index defined in Eqn. 4.4 [42] – [43].

$$SSIM(A,B) = \frac{(2\mu_A\mu_B + C_1)(2\sigma_A\sigma_B + C_2)}{(\mu_A{}^2 + \mu_B{}^2 + C_1)(\sigma_A{}^2 + \sigma_A{}^2 + C_2)} \tag{4.4}$$

Where,

μ_A, μ_B, = mean intensities of original data A and reconstructed data B;

σ_A, σ_B = standard deviation of original data A and reconstructed data B;

C_1, C_2 = constant. The standard value of C1 and C2 are 0.01 and 0.03 respectively [42].

(a) SSIM 0.9921

(b) SSIM 0.9979

(c) SSIM 0.9994

(d) SSIM 1

Figure 4-2 Reconstructed image SSIM index with a number of extra bits in hardware 0, 1, 2, 3 bits in Figure a, b, c, and d respectively, for the proposed algorithm $YC_MC_EC_F$.

The reconstructed image data can be achieved exactly as the original data for the lossless compression. If so, the SSIM index will be 1. The proposed algorithm $YC_MC_EC_F$ is shown with the SSIM index comparison in Figure 4-2, with the reconstructed image using 0, 1, 2, 3 bits in hardware.

4.3 Performance evaluation

Performance evaluation is computed for the lossless proposed compression design. The proposed compression design is applied on the natural image and wireless capsule endoscopic image. In the proposed algorithm, the compression design is realized with both $YC_MC_EC_F$ and $YC_DC_MC_O$ color transformations. The color transformations have been proposed to obtain the highest compression from the Bayer CFA images. In most of the research work, the CFA images are generated from

the full color images, which are analyzed after processing. The analysis does not provide the actual evaluation of the Bayer CFA images. Therefore, the proposed compression also uses the real CFA images.

The results are compared with real CFA images and generated CFA images for the proposed methods. The performance evaluation of the proposed work has been carried out using the MATLAB tool. The standard dataset for the generated CFA images are KODAK (24 sample images) [44], Nikon D90 high resolution (6 sample images), and Olympus E-P1 (24 sample images). The real CFA images (14 samples) are evaluated in the proposed work. The purpose of the performance assessment of the compression algorithm was to measure the compression ratio. The high compression ratio is important parameter to save transmission power.

4.3.1 Natural image

The compression method 1 is developed with the $YC_MC_EC_F$ color transformation while the $YC_DC_MC_O$ color transformation is used in compression method 2. Method 2 outperforms method 1 in terms of compression. The color transformation considering the minimum value of the correlation among the components, the standard deviation and entropy of the components and the efficient encoding of the prediction model provides the best compression design among the recently developed compression algorithms, such as JPEG-LS [21], JPEG2000 [22], LCMI [15], CMBP [9], and HP [3].

The kodak is popular dataset for CFA images. Several comparisons have been carried out with the kodak dataset. The proposed method is compared with common JPEG-LS, JPEG2000, LCMI, CMBP, and HP. With the kodak dataset, the proposed compression method 2 provides 56.91%, 33.24%, 30.05%, 22.87%, 20.74%, and 4.52% more compression than JPEG-LS, JPEG2000, LCMI, CMBP, and HP, respectively. The results, shown in Table 4-1, result shows that the best compression can be achieved for the kodak dataset from the proposed method 2 with the $YC_DC_MC_O$ color transformation.

Table 4-1: Bit rate of different compression algorithms for KODAK dataset in BPP.

KODAK	JPEG-LS [21]	JPEG 2K [22]	LCMI [15]	CMBP [9]	HP [3]	YLMN [13]	$YC_MC_EC_F$	$YC_DC_MC_O$
1	6.40	5.81	5.82	5.48	5.44	6.86	**4.97**	**4.88**
2	6.79	5.13	4.63	4.33	4.27	5.60	**3.59**	**3.36**
3	5.88	4.22	3.97	3.75	3.68	4.95	**3.06**	**2.84**
4	6.68	4.93	4.61	4.38	4.36	5.62	**3.63**	**3.45**
5	6.47	5.95	5.86	5.41	5.34	7.06	**5.13**	**4.97**
6	5.87	5.21	5.14	4.88	4.79	5.84	**3.91**	**3.74**
7	5.97	4.50	4.30	3.96	3.84	5.31	**3.32**	**3.16**
8	6.30	5.90	5.97	5.57	5.51	7.15	**5.21**	**5.21**
9	5.07	4.39	4.32	4.19	4.12	5.41	**3.37**	**3.22**
10	5.40	4.56	4.42	4.23	4.16	5.56	**3.50**	**3.35**
11	5.37	4.99	4.95	4.68	4.63	5.88	**3.92**	**3.77**
12	5.63	4.49	4.31	4.09	4.01	5.19	**3.19**	**2.99**
13	6.75	6.37	6.50	6.14	6.07	7.22	**5.35**	**5.17**
14	6.29	5.56	5.49	5.17	5.10	6.40	**4.43**	**4.25**
15	6.32	4.66	4.40	4.10	3.98	5.56	**3.62**	**3.43**
16	5.29	4.55	4.52	4.38	4.31	5.25	**3.29**	**3.12**
17	4.97	4.55	4.50	4.29	4.23	5.65	**3.63**	**3.45**
18	6.18	5.57	5.54	5.28	5.24	6.50	**4.55**	**4.36**
19	5.47	4.91	4.90	4.71	4.65	5.96	**3.98**	**3.84**
20	4.32	4.03	4.05	3.54	3.19	4.67	**3.13**	**2.91**
21	5.47	5.04	4.98	4.80	4.70	5.74	**3.74**	**3.56**
22	6.19	5.22	5.06	4.85	4.80	6.10	**4.15**	**3.96**
23	6.83	4.53	3.96	3.85	3.76	5.09	**3.13**	**2.86**
24	5.72	5.22	5.26	4.87	4.75	6.49	**4.51**	**4.41**
Avg.	5.90	5.01	4.89	4.62	4.54	5.88	**3.93**	**3.76**

Table 4-2: Comparison of bit rate of different compression algorithms for Olympus E-P1 dataset in BPP

E-P1	JPEG-LS [21]	JPEG 2K [22]	LCMI [15]	CMBP [9]	HP [3]	YLMN [13]	$YC_MC_EC_F$	$YC_DC_MC_O$
1	7.25	5.28	4.63	4.39	4.25	5.96	**3.89**	**3.75**
2	5.93	4.61	4.16	4.06	3.89	5.55	**3.44**	**3.40**
3	7.24	5.26	4.58	4.53	4.35	5.97	**4.00**	**3.68**
4	7.24	5.04	4.29	4.24	4.06	5.63	**3.73**	**3.40**
5	7.15	4.58	3.43	3.38	3.17	4.71	**2.80**	**2.57**
6	6.23	4.92	4.53	4.39	4.26	5.89	**3.72**	**3.61**
7	7.32	4.45	3.28	3.32	3.12	4.52	**2.65**	**2.38**
8	7.47	5.73	5.23	5.06	4.93	6.62	**4.63**	**4.53**
9	6.80	4.79	4.00	3.94	3.76	5.42	**3.37**	**3.14**
10	5.93	4.57	4.28	4.19	4.06	5.70	**3.56**	**3.52**
11	5.47	3.96	3.60	3.50	3.35	4.65	**2.81**	**2.75**
12	4.60	3.30	3.04	3.00	2.83	3.88	**2.22**	**2.13**
13	6.85	4.96	4.37	4.38	4.28	5.51	**3.43**	**3.29**
14	6.01	4.52	4.21	4.18	4.08	5.35	**3.34**	**3.22**
15	6.19	4.53	4.14	4.17	4.05	5.31	**3.22**	**3.09**
16	6.94	4.91	4.28	4.19	4.20	5.43	**3.35**	**3.17**
17	6.36	4.10	3.22	3.07	2.89	4.23	**2.40**	**2.23**
18	6.81	4.30	3.48	3.25	3.05	4.50	**2.65**	**2.46**
19	4.74	3.26	2.98	2.93	2.76	3.87	**2.07**	**1.94**
20	5.00	3.42	3.14	3.03	2.85	4.04	**2.27**	**2.15**
21	4.73	3.07	2.79	2.74	2.55	3.51	**1.92**	**1.79**
22	4.59	3.30	3.07	2.97	2.79	4.01	**2.23**	**2.08**
23	4.54	3.91	3.84	3.63	3.46	4.92	**3.00**	**2.87**
24	5.45	3.77	3.37	3.19	2.99	4.35	**2.52**	**2.37**
Avg.	6.12	4.36	3.83	3.74	3.58	4.98	**3.05**	**2.90**

Table 4-3: Comparison of bit rate of different compression algorithms for REAL CFA images in BPP.

R CFA	JPEG-LS [21]	JPEG 2K [22]	LCMI [15]	CMBP [9]	HP [3]	YLMN [13]	$YC_MC_EC_F$	$YC_DC_MC_O$
1	6.42	5.39	5.26	5.00	4.90	6.10	**4.13**	**3.88**
2	6.42	5.49	5.33	5.09	4.99	6.15	**4.18**	**3.96**
3	6.56	5.39	5.23	5.00	4.90	6.07	**4.11**	**3.88**
4	5.54	4.55	4.49	4.26	4.13	5.31	**3.31**	**3.05**
5	5.52	4.45	4.39	4.17	4.04	5.20	**3.21**	**2.96**
6	5.23	4.61	4.55	4.39	4.09	5.37	**3.40**	**3.19**
7	4.81	3.81	3.70	3.61	3.36	4.49	**2.75**	**2.52**
8	4.77	3.72	3.62	3.54	3.28	4.41	**2.69**	**2.47**
9	5.50	4.74	4.71	4.47	4.27	5.53	**3.72**	**3.50**
10	5.56	4.73	4.70	4.46	4.26	5.51	**3.71**	**3.49**
11	4.85	3.97	3.95	3.74	3.53	4.72	**2.91**	**2.70**
12	4.78	3.92	3.91	3.70	3.49	4.68	**2.89**	**2.68**
13	4.61	3.86	3.87	3.68	3.45	4.68	**2.88**	**2.66**
14	4.87	4.11	4.11	3.91	3.69	4.92	**3.13**	**2.91**
Avg.	5.39	4.48	4.42	4.22	4.03	5.23	**3.36**	**3.13**

Table 4-4: Comparison of different compression algorithms for the D90 dataset in BPP.

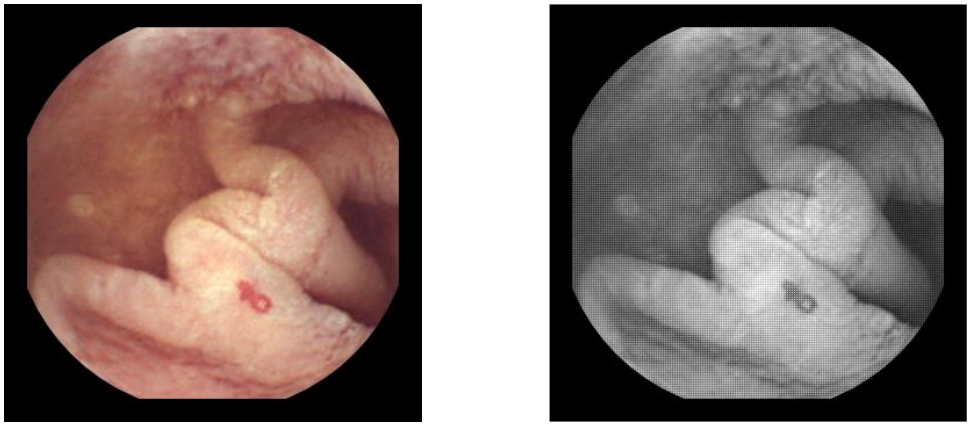
D 90	JPEG-LS [21]	JPEG 2K [22]	LCMI [15]	CMBP [9]	HP [3]	YLMN [13]	$YC_MC_EC_F$	$YC_DC_MC_O$
1	4.93	3.95	3.68	3.66	3.54	4.74	**2.71**	**2.51**
2	6.23	5.21	4.95	4.67	4.51	6.19	**4.10**	**3.98**
3	4.87	3.82	3.69	3.52	3.40	4.69	**2.89**	**2.76**
4	5.30	3.50	3.11	3.09	2.93	3.93	**2.01**	**1.95**
5	5.00	3.84	3.49	3.35	3.18	4.45	**2.67**	**2.61**

6	4.40	3.35	2.94	2.84	2.67	3.45	**2.02**	**1.88**
Avg.	5.12	3.95	3.64	3.52	3.37	4.57	**2.73**	**2.62**

The real CFA images are used to evaluate the performance of the compression algorithms. As the raw CFA images do not have any pre-or post-processing, the actual compression performance can be achieved by comparing with the state of the art methods. Fourteen real CFA images with a resolution of 1024 x 768 are used for compression. The proposed method 2 requires the least bits per pixel. It uses 72.2%, 43.13%, 41.21%, 34.82%, 28.75% and 7.35% less bits than JPEG-LS, JPEG-2K, LCMI, CMBP, HP, and proposed method 1, respectively. The compression result of the real CFA images is shown in Table 4-3. The proposed method 2 also outperforms the other methods for the real CFA images. The proposed algorithm is compared for NIKON D90 (4288 x 2848) and OLYMPUS E-P1 (4032 x 3024) high resolution datasets. The proposed algorithm uses only 2.62 bits per pixel for the NIKON D90 dataset and 2.90 bits per pixel for the OLYMPUS E-P1 dataset, which are the lowest bits per pixel in the state of the arts. The results are compared in Table 4-4 and Table 4-2 for D90 and E-P1 datasets, respectively.

4.3.2 WCE image

The proposed algorithm is realized with 20 endoscopic images of different parts of the gastrointestinal tract. The original WCE and CFA WCE images are shown in Figure 4-3. The performances of different image compression algorithms are compared in Table 4-5.



(a) (b)

Figure 4-3: (a) original WCE image and (b) CFA WCE images

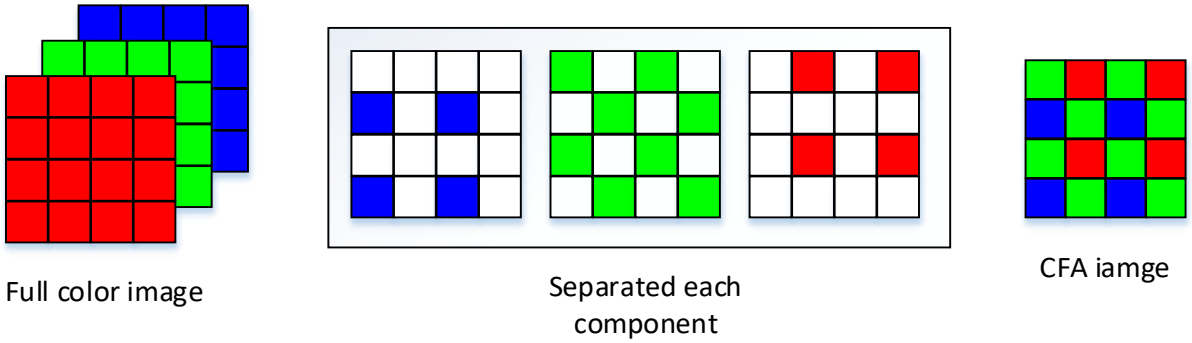
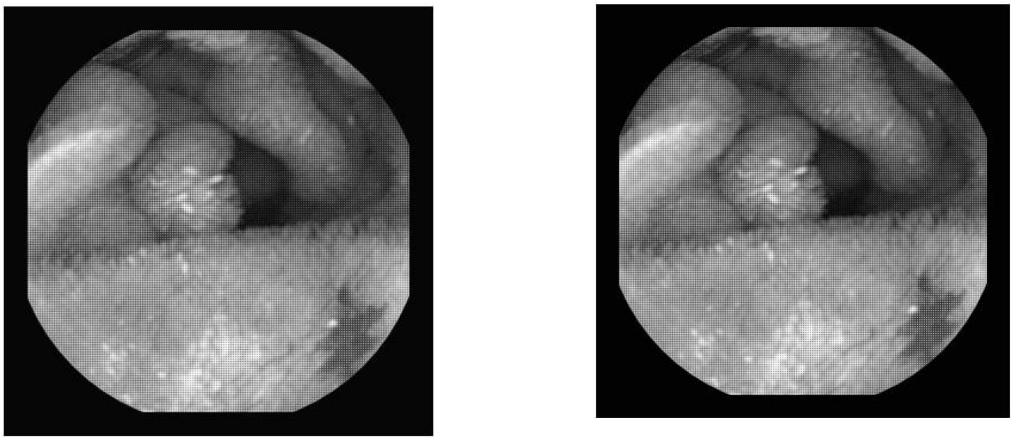


Figure 4-4 Simulated CFA image generation

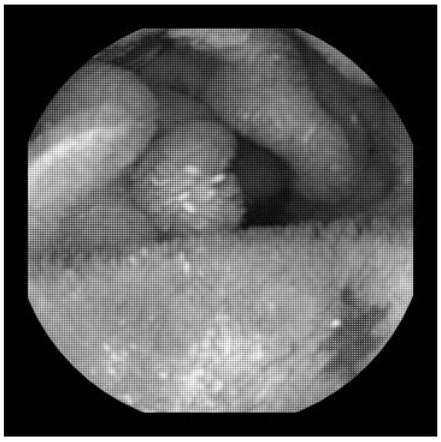
Figure 4-4 shows the process of generating a simulated WCE image from the RGB image. The proposed image compression provides two new color transformations for the Bayer CFA image. The motivation for the color transformation is to generate fewer correlated color components. On the other hand, WCE images have demonstrated inter pixel correlation [30] – [39]. Therefore, the proposed color transformations work more efficiently in WCE images.

4.3.2.1 Performance parameters PSNR and SSIM

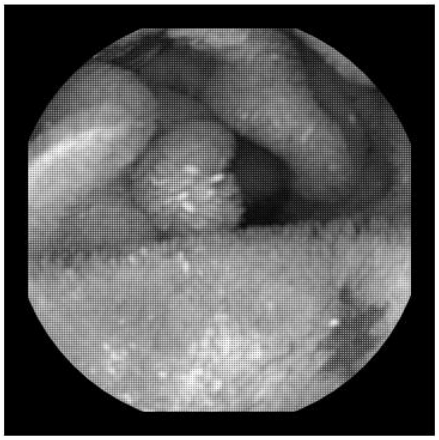


(a) PSNR = 42.63, SSIM = 0.9689

(b) PSNR = 48.50, SSIM = 0.9719



(c) PSNR = 54.18, SSIM = 0.9840



(d) PSNR = Inf, SSIM = 1

Figure 4-5: The WCE image performance of a reconstructed image using 0, 1, 2, and 3 extra bits in the hardware a, b, c, and d, respectively.

The compression ratio for the WCE image is high in the state of the art. The performance parameter for the proposed near lossless image compression is shown in Figure 4-5. The number of extra bits in the hardware design gives better PSNR and SSIM indexes of the reconstructed image.

4.3.2.2 Performance of compression

The number bits per pixel is compared in Table 9 for lossless image compression. The performance of Bayer RGB color space in terms of compression is poor compared to all the other compression algorithms. The color components of Bayer RGB are not suitable for compression-first method as previously discussed.

The YLMN compression algorithm is efficient in hardware design but it has a lower compression ratio than the proposed algorithm. The YDgCoCg compression algorithm performs better than the YLMN compression. The proposed $YC_MC_EC_F$ and $YC_DC_MC_O$ compression algorithms use only 2.55 BPP and 2.39 BPP respectively. It is showed that the performance of the proposed algorithms is better in the state of the art studies.

Table 4-5: Bit rate for different WCE image compression algorithms for KID dataset in BPP.

	YDgCoCg [16]	YLMN [13]	$YC_MC_EC_F$	$YC_DC_MC_O$
1	4.21	4.25	2.62	**2.45**
2	4.07	4.13	2.50	**2.32**
3	4.06	4.09	2.48	**2.33**
4	4.10	4.17	2.53	**2.42**
5	4.26	4.29	2.64	**2.49**
6	4.26	4.31	2.67	**2.50**
7	4.19	4.23	2.60	**2.44**
8	4.38	4.41	2.75	**2.60**
9	4.04	4.06	2.45	**2.30**
10	4.29	4.31	2.71	**2.53**
11	3.84	3.88	2.34	**2.18**
12	4.03	4.05	2.47	**2.29**
13	4.04	4.09	2.50	**2.33**
14	3.97	4.02	2.43	**2.25**
15	4.10	4.11	2.52	**2.36**
16	4.07	4.09	2.48	**2.33**
17	3.96	4.00	2.43	**2.27**
18	4.14	4.18	2.61	**2.40**
19	4.25	4.27	2.64	**2.48**
20	4.22	4.26	2.65	**2.46**
AVG	4.12	4.16	2.55	**2.39**

4.4 Summary

The performance of the proposed compression algorithm has been compared for both the natural image and the WCE image. This algorithm outperforms existing compression algorithm in terms of compression. The comparison is shown using the popular KODAK, D90, Olympus-EP1, real CFA for the natural image and the KID dataset for the WCE image [45].

Chapter 5 HARDWARE IMPLEMENTATION

5.1 Introduction

The proposed algorithm is implemented in Verilog HDL language and simulated for functional verification. The design is synthesized with TSMC 65 nm CMOS technology. The important module of the design contains the color transformation module, prediction module and encoding module. The design is implemented in FPGA board. The design is implemented for the 640x640 resolution, 8 bits per pixel Bayer images. The Block diagram of the hardware architecture is shown in Figure 5-1. The design is synthesized in Synopsys EDA tool, Design Compiler. The gate counts, area and power consumption are compared in Table 9.

5.2 Hardware architecture

The image pixels store in a file. Each pixel value is sent to the image compressor for processing. Hardware design model consists of six essential modules such as Color transformation module, Mux/De-Mux, Memory, Prediction module, Control unit and Encoder. The whole architecture is shown in Figure 5-1.

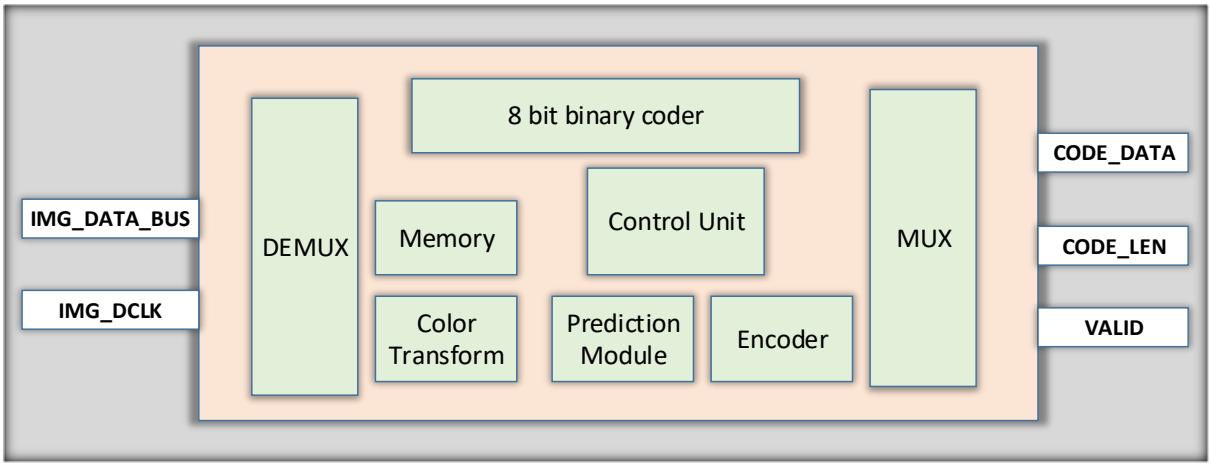


Figure 5-1: Hardware Architecture of the Compression

5.2.1 Color transformation module

The color transformation module converts the color plane into the desired color plane. In this work, two color conversions are implemented in two different designs. The color conversions are

executed using the equations described in section 3.1.2 and 3.1.3. The color components are converted from Bayer RGB pixels to $YC_MC_EC_F$ and $YC_DC_MC_O$.

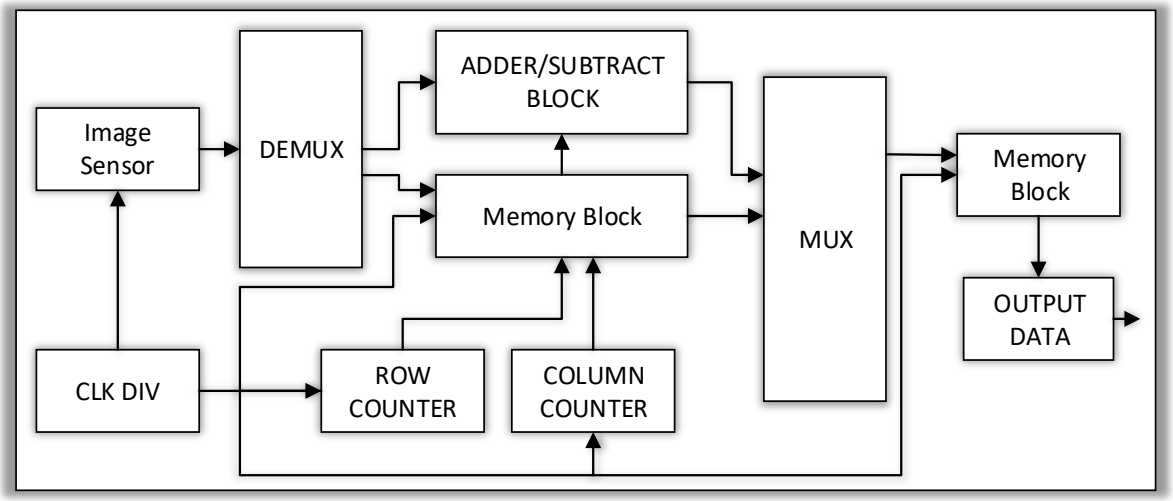


Figure 5-2: Block diagram of color transformation module

At each clock cycle one pixel is sent to the memory. When the second row of 2x2 block pixel starts then it converts the Bayer RGB four components to new color transformation using the pixel from the memory. After the execution of first block then it moves to next 2x2 block unit. The block diagram of the color transformation module is shown in Figure 5-2. Full image is transformed into $YC_MC_EC_F$ and $YC_DC_MC_O$ color planes.

5.2.2 Multiplexer/De-Multiplexer

Multiplexer and de-multiplexer are used before and after the transformation unit. They control the access of the pixel entry to the transform unit. Multiplexer decides the pixel position status and send it to the right component of the design. If the pixel is in the first row, then it is being stored in the memory and if it is in the second row then after getting each odd pixel of the second row it executes the transformation. Therefore, the multiplexer is doing all the selection which module should be executed according to the pixel position.

5.2.3 Memory

Memory unit is essential part of the design for storing the first row of the image pixels. When it starts getting the second row of the image then it passes the 4x4 block to the transformation unit. The memory is synthesized in the Synopsys dc_compiler. It also measures the power, frequency and area of the design.

5.2.4 Prediction module

The transformation unit sends the converted pixel value to the prediction unit. If it is the first pixel value, then the prediction module sends directly to the output of the value and stores the pixel value in a register. When the second pixel gets into the prediction module then it sends the difference between the two pixels' value to the encoder module. Before it sends the difference value to the encoder, it does the mapping for the positive difference as even number by using the Eqn. 16 and negative difference value as positive odd number by using the Eqn. 17.

5.2.5 Control unit

Control unit is essential part of the whole design. It generates all the control signal of the compression algorithm. It generates the signal using combinational signals and it receives the clock, reset and enable as input of the unit. It controls the DEMUX output in different state of the design. And it generates signal for the MUX selection to give the correct output.

5.2.6 Encoder module

The Encoder module receives the DPCM output data and mapped the received data using the Eqn. 16 and Eqn. 17 such that the negative data as odd and positive data as even. Then it uses the Golomb Rice Encoding to generate the output bit stream. The Encoder is built with shifter as it needs to be divided by the power of 2. It has the simple logical components in the design.

5.3 Results and comparison

The hardware is synthesized in the Synopsys DC compiler. It also simulated in the ModelSim-Altera (10.1b) for the functional verification. Then we used the vcd file from the ModelSim to simulate the design in Quartus II (13.0sp1). The Cyclone IV FPGA family is selected for the

simulation. The maximum frequency of the design was 250MHz. The area, gate counts and power are calculated. The comparisons are shown with different design in Table 5-1.

5.3.1 Hardware cost

The critical part of the hardware design was to save the area and power. Synopsys DC compiler was used to synthesize the design and the frequency of the design was 250MHz. In comparison to the design of the others, the area and power are better than all of them except for the YLMN compression design [13] and trade-off is the result of the compression. Therefore, low complexity and better compression has been achieved in the proposed compression algorithm. It uses 640 x 640 images where the power is very low.

The power measurement is processed with 250 MHz frequency. The system consumes 1.74 mW power. The design consists of $YC_MC_EC_F$ color transformation, simple prediction model, easy implementation encoder, therefore in terms of power and area it outperforms the other works with better compression result.

Table 5-1: Hardware comparison of different compression algorithms for Bayer CFA images

	[34]	[36]	[37]	[41]	[13]	Proposed
Process(nm)	180	180	180	180	65	65
Frequency(MHz)	20-24	20	200	200	250	250
Gate Counts (k)	19.5	50	11.57	5.54	3.78	5.2
Memory(k)	17.5	29	15.2	10.2	2.93	5.74
Normalized Area	3.75	9.62	2.92	1.07	0.73	1

5.4 Summary

The performance of the hardware design is compared in terms of gate counts, memory, area and power. The proposed algorithm outperforms all the existing compression algorithm except for the YLMN compression algorithm. Although the compression ratio for the YLMN is less than the

proposed algorithms. The tradeoff of using the less resource usage is the compression performance. On the other hand, the proposed algorithms show the best compression performance using the optimize resource usages. In consideration of the performance and optimize resource usages the proposed compression algorithms are the best CFA compression algorithm in the state of the art.

Chapter 6 CONCLUSION AND FUTURE WORK

6.1 Conclusion

In this work, a low complexity lossless compression algorithm is proposed along with two new color transformations for the Bayer pattern CFA image. The proposed color transformation offers de-correlated color components for better compression. The color components only store the necessary information discarding the repeated information in the other components. The features of the color transformation are evaluated in the Chapter 3. The dynamic intensity, standard deviation, entropy and prediction gain indicates that the proposed color transformation is efficient color transformation compared to the Bayer RGB, YC_bC_r, $YD_gC_oC_g$, and YLMN color transformation.

The prediction gain for the DPCM model is calculated for the different color transformation which also gives better result for the proposed color transformation. Therefore, it is clear that the compression ratio will be better for proposed compression algorithm. Later in chapter 4 the results are compared among the different compression algorithm. The proposed algorithms are realized with simulated CFA image and real CFA image. The proposed method outperforms all the existing state of the arts algorithms. The average bit per pixel value of the Kodak dataset is 3.93 and 3.76 for the $YC_MC_EC_F$ and $YC_DC_MC_O$ respectively.

In terms of the complexity, the proposed method uses DPCM prediction model, which is simpler than existing context models or existing prediction models. The adaptive Golomb Rice encoder is straightforward to implement. The hardware implementation cost is optimized for the resource usage. Although YLMN compression algorithm provides less implementation cost. It uses less memory than the proposed algorithm, therefore the area and power cost is less than the proposed algorithm. But YLMN is less efficient compared to the proposed compression algorithm in terms of compression ratio. The features of proposed color transformations are also efficient in the WCE image. The result shows that the proposed compression algorithm is also efficient for WCE system. Finally, the proposed algorithm with new two-color transformations provides better result and optimize hardware implementation cost than existing works.

6.2 Future work

The following extensions of this work can be considered as future work:

- The proposed algorithm uses DPCM prediction model for the simple implementation of the design. The performance of the compression can be improved further using a complex prediction model or different prediction model.
- The adaptive Golomb rice is an efficient encoding method. Different encoding strategy can be applied to get better result.
- The compression algorithm is considered to reduce the redundancy in a single frame. The consecutive frames contain more redundant information therefore better compression can be achieved reducing the frame to frame information. The color transformation can be used in the video compression.
- The real-time prototype implementation can be developed for the wireless endoscopic image. The prototype can be used to see how the compression performs in the real world.
- The hardware implementation can be improved. The proposed algorithm is using the optimize resources with the better compression ratio. The performance can be compromised to reduce the implementation cost. The power and area is more important than performance in some application.

REFERENCES

[1] Bayer, Bryce E. "Color imaging array." U.S. Patent No. 3,971,065. 20 Jul.1976.

[2] G. Iddan, G. Meron, A. Glukhovsky, and P. Swain, “Wireless capsule endoscopy,” *Nature*, vol. 405, no. 6785, p. 417, 2000.

[3] S. Kim and N. I. Cho, "Lossless Compression of Color Filter Array Images by Hierarchical Prediction and Context Modeling," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 24, no. 6, pp. 1040-1046, 2014.

[4] C. C. Koh, J. Mukherjee and S. K. Mitra, "New efficient methods of image compression in digital cameras with color filter array," *IEEE Transactions on Consumer Electronics*, vol. 49, no. 4, pp. 1448-1456, 2003.

[5] K. Nallaperumal, S. Christopher, S. S. Vinsley, and R. K. Selvakumar. "New efficient image compression method for single sensor digital camera images." *Conference on Computational Intelligence and Multimedia Applications, 2007. IEEE International Conference on 2007*, vol. 3, pp. 113-117.

[6] X. Xie, G. Li, Z. Wang, C. Zhang, D. Li, X. Li. “A new method of lossy image compression for digital image sensors with Bayer color filter arrays.” *Circuits and Systems, 2005. ISCAS 2005. IEEE International Symposium on May 2005*, pp. 4995-4998.

[7] A. Bazhyna and K. Egiazarian, "Lossless and near lossless compression of real color filter array data," *IEEE Transactions on Consumer Electronics*, vol. 54, no. 4, pp. 1492-1500, 2008.

[8] K. H. Chung, Y. H. Chan. “A fast-reversible compression algorithm for Bayer color filter array images.” *In Proceedings: APSIPA ASC 2009: Asia-Pacific Signal and Information Processing Association, 2009 Annual Summit and Conference 2009 Oct 4*, pp. 825-828.

[9] K. H. Chung and Y. H. Chan, "A Lossless Compression Scheme for Bayer Color Filter Array Images," *IEEE Transactions on Image Processing*, vol. 17, no. 2, pp. 134-144, 2008.

[10] S. Y. Lee, and A. Ortega. "A New Approach for Compression of Images Captured using Bayer Color Filter Arrays." *arXiv preprint arXiv: 0903.2272*, 2009.

[11] S. Y. Lee, A. Ortega. “A novel approach of image compression in digital cameras with a Bayer color filter array.” *IEEE Image Processing, 2001. Proceedings. 2001 International Conference on 2001* Vol. 3, pp. 482-485.

[12] N. Zhang, X. Wu. “Lossless compression of color mosaic images.” *IEEE Transactions on Image Processing*, vol. 15, no. 6, pp. 1379-1388, 2006.

[13] S. K. Mohammed, K. M. M. Rahman, and K. A. Wahid. "Lossless Compression in Bayer Color Filter Array for Capsule Endoscopy." *IEEE Access*, vol. 5, pp. 13823-13834, 2017.

[14] S. L. Chen, Y. R. Chen, T. L. Lin, and Z. Y. Liu. "A cost-efficient lossless compression color filter array images VLSI design for wireless capsule endoscopy." *Journal of Medical Imaging and Health Informatics*, vol. 5, no. 2, pp. 378-384, 2015.

[15] M. Yang, and N. Bourbakis. "An overview of lossless digital image compression techniques." *Circuits and systems, 2005. 48th Midwest symposium on 2005,* pp. 1099-1102.

[16] H. S. Malvar, G. J. Sullivan. "Progressive-to-lossless compression of color-filter-array images using macropixel spectral-spatial transformation." *Data Compression Conference (DCC) on April 2012,* pp. 3-12.

[17] D. Lee, and K. N. Plataniotis. "Lossless compression of HDR color filter array image for the digital camera pipeline." *Signal Processing: Image Communication*, vol. 27, no. 6, pp. 637-649, 2012.

[18] X. Xie, G. Li, X. Li, Z. Wang, C. Zhang, D. Li, L. Zhang. "A new approach for near-lossless and lossless image compression with Bayer color filter arrays." *IEEE Image and Graphics (ICIG'04), Third International Conference on Dec 2004,* pp. 357-360.

[19] M. Wang, B. T. "Generalized YUV demosaicking of CFA images." *Image Processing (ICIP), 2010 17th IEEE International Conference on Sep 2010,* pp. 1909-1912.

[20] G. K. Wallace. "The JPEG still picture compression standard." *IEEE transactions on consumer electronics*, vol. 38, no. 1, pp. xviii-xxxiv, 1992.

[21] M. J. Weinberger, G. Seroussi, and G. Sapiro. "The LOCO-I lossless image compression algorithm: Principles and standardization into JPEG-LS." *IEEE Transactions on Image processing* vol. 9, no. 8, pp. 1309-1324, 2000.

[22] A. Skodras, C. Christopoulos, and T. Ebrahimi. "The JPEG 2000 still image compression standard." *IEEE Signal processing magazine* vol. 18, no. 5, pp. 36-58, 2001.

[23] G. K. Wallace. "The JPEG Still Picture Compression Standard Communication of the ACM 34.", pp. 3-44, 1991.

[24] H. S. Malvar, A. Hallapuro, M. Karczewicz, and L. Kerofsky. "Low-complexity transform and quantization in H. 264/AVC." *IEEE Transactions on circuits and systems for video technology*, vol. 13, no. 7, pp. 598-603, 2003.

[25] T. H. Khan, and K. Wahid. "Low-complexity colour-space for capsule endoscopy image compression." *Electronics letters* vol. 47, no. 22, pp. 1217-1218, 2011.

[26] T. H. Khan, and K. A. Wahid. "Design of a lossless image compression system for video capsule endoscopy and its performance in in-vivo trials." *Sensors* vol. 14, no. 11, pp. 20779-20799, 2014.

[27] H. S. Malvar, G. J. Sullivan, and S. Srinivasan. "Lifting-based reversible color transformations for image compression." *Applications of Digital Image Processing XXXI 2008*, Vol. 7073, p. 707307.

[28] S. L. Chen, Y. R. Chen, T. L. Lin, Z. Y. Liu. "A cost-efficient lossless compression color filter array images VLSI design for wireless capsule endoscopy." *Journal of Medical Imaging and Health Informatics.* Vol. 5, no. 2, pp. 378-84, 2015.

[29] A. Menciassi, C. Gastone, and C. Cavallotti. "Future developments of video capsule endoscopy: Hardware." *Video Capsule Endoscopy. Springer, Berlin, Heidelberg,* pp. 543-556, 2014.

[30] A. Karargyris, and N. Bourbakis. "Wireless capsule endoscopy and endoscopic imaging: A survey on various methodologies presented." *IEEE Engineering in medicine and biology magazine* vol. 29, no. 1, pp. 72-83, 2010.

[31] A. M. Singeap, C. Stanciu, and A. Trifan. "Capsule endoscopy: The road ahead." *World journal of gastroenterology*, vol. 22, no. 1, pp. 369, 2016.

[32] A. J. Barkin, and J. S. Barkin. "Video Capsule Endoscopy: Technology, Reading, and Troubleshooting." *Gastrointestinal Endoscopy Clinics,* vol. 27, no. 1, pp. 15-27, 2017.

[33] P. Turcza, and M. Duplaga. "Hardware-efficient low-power image processing system for wireless capsule endoscopy." *IEEE journal of biomedical and health informatics,* vol. 17, no. 6, pp. 1046-1056, 2013.

[34] X. Chen, X. Zhang, L. Zhang, X. Li, N. Qi, H. Jiang, and Z. Wang. "A wireless capsule endoscope system with low-power controlling and processing ASIC." *IEEE Transactions on Biomedical Circuits and Systems*, vol. 3, no. 1, pp. 11-22, 2009.

[35] K. Wahid, S. B. Ko, D. Teng. "Efficient hardware implementation of an image compressor for wireless capsule endoscopy applications." *Neural Networks, 2008. IJCNN 2008. (IEEE World Congress on Computational Intelligence). IEEE International Joint Conference on Jun 2008*, pp. 2761-2765.

[36] X. Chen, H. Jiang, X. Li, and Z. Wang. "A new compression method for wireless image sensor node." *Solid-State Circuits Conference. ASSCC'07. IEEE Asian on 2007*, pp. 184-187.

[37] T. H. Tsai, Y. H. Lee and Y. Y. Lee, "Design and Analysis of High-Throughput Lossless Image Compression Engine Using VLSI-Oriented FELICS Algorithm," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 18, no. 1, pp. 39-52, 2010.

[38] K. A. Fante, B. Bhaumik, and Sh. Chatterjee. "Design and implementation of computationally efficient image compressor for wireless capsule endoscopy." *Circuits, Systems, and Signal Processing* vol. 35, no. 5, pp. 1677-1703, 2016.

[39] P. Turcza, M. Duplaga. "Low-power image compression for wireless capsule endoscopy." *Imaging Systems and Techniques, 2007. IST'07. IEEE International Workshop on May 2007*, pp. 1-4.

[40] S. E. Ghrare, M. A. Ali, M. Ismail, K. Jumari. "Diagnostic quality of compressed medical images: Objective and subjective evaluation." *Modeling & Simulation, 2008. AICMS 08. Second Asia IEEE International Conference on May 2008,* pp. 923-927.

[41] S. L. Chen, Y. R. Chen, T. L. Lin, & Z. Y. Liu. "A cost-efficient lossless compression color filter array images VLSI design for wireless capsule endoscopy." *Journal of Medical Imaging and Health Informatics,* vol. 5, no. 2, pp. 378-384, 2015.

[42] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli. "Image quality assessment: from error visibility to structural similarity." *IEEE transactions on image processing*, vol. 13, no. 4, pp. 600-612, 2004.

[43] Z. Wang, A. C. Bovik, and H. R. Sheikh. "Structural similarity based image quality assessment." *Digital Video image quality and perceptual coding,* pp. 225-241, 2005.

[44] R. Franzen. (1999). Kodak lossless true color image suite. source: http://r0k. us/graphics/kodak, 4.

[45] D. K. Koulaouzidis, A. & Iakovidis, "KID, a capsule endoscopy database for medical decision support," 2016. [Online]. Available: http://is-innovation.eu/kid.

[46] S. Kak. "Generalized unary coding." *Circuits, Systems, and Signal Processing*, vol. 35, no. 4, pp. 1419-1426, 2016.

[47] G. Sharma, and H. J. Trussell. "Digital color imaging." *IEEE transactions on image processing*, vol. 6. no. 7, pp. 901-932, 1997.

[48] R. Yu, C. C. Ko, S. Rahardja, X. Lin. “Bit-plane Golomb coding for sources with Laplacian distributions.” *Acoustics, Speech, and Signal Processing, 2003. Proceedings. (ICASSP'03). 2003 IEEE International Conference on April 2003,* Vol. 4, pp. IV-277.

[49] S. Golomb. “Run-length encodings (Corresp.).” *IEEE transactions on information theory.* Vol. 12, no. 3, pp. 399-401, 1966.

[50] J. Ziv and A. Lempel, "A universal algorithm for sequential data compression,*" IEEE Transactions on Information Theory,* vol. 23, no. 3, pp. 337 - 343, 1977.

[51] R. Tajallipour and K. Wahid, "Efficient data encoder for low-power capsule endoscopy application," *Int. Conf. on Information Sciences Signal Processing and their Applications, 2010,* pp. 512-515.

[52] D. Turgis and R. Puers, "Image compression in video radio transmission for capsule endoscopy," *Elsevier Sensors and Actuators A: Physical*, vol 123 pp. 129 - 136, 2005.

[53] S. Rigler, W. Bishop and A. Kennings, "FPGA-based lossless data compression using Huffman and LZ77 algorithms," *IEEE Canadian Conf. on Elec. and Comp. Engin., 2007,* pp. 1235-1238.

[54] C. Hu, M. Meng, L. Liu, Y. Pan and Z. Liu, "Image representation and compression for capsule endoscope robot," *IEEE International Conference on Information and Automation, 2009,* pp. 506-511.

[55] M. A. Podpora. ’YUV vs. RGB–A comparison of lossy compressions for human-oriented man-machine interfaces’. *III SWD conference proceedings, Glucholazy*. pp. 1429-1533.

APPENDIX A

List of publications

A.1 Published journals

[1] S. K. Mohammed, **K. M. M. Rahman**, and K. A. Wahid. "Lossless Compression in Bayer Color Filter Array for Capsule Endoscopy." *IEEE Access* vol. 5, pp.13823-13834, 2017.

A.2 Submitted journals

[1] **K M Mafijur Rahman**, and Shahed K. Mohammed, Khan A. Wahid. “A low complexity lossless Bayer color filter array image compression algorithm.” *IEEE transactions on consumer electronics.*

APPENDIX B

Figure B-1: KODAK dataset

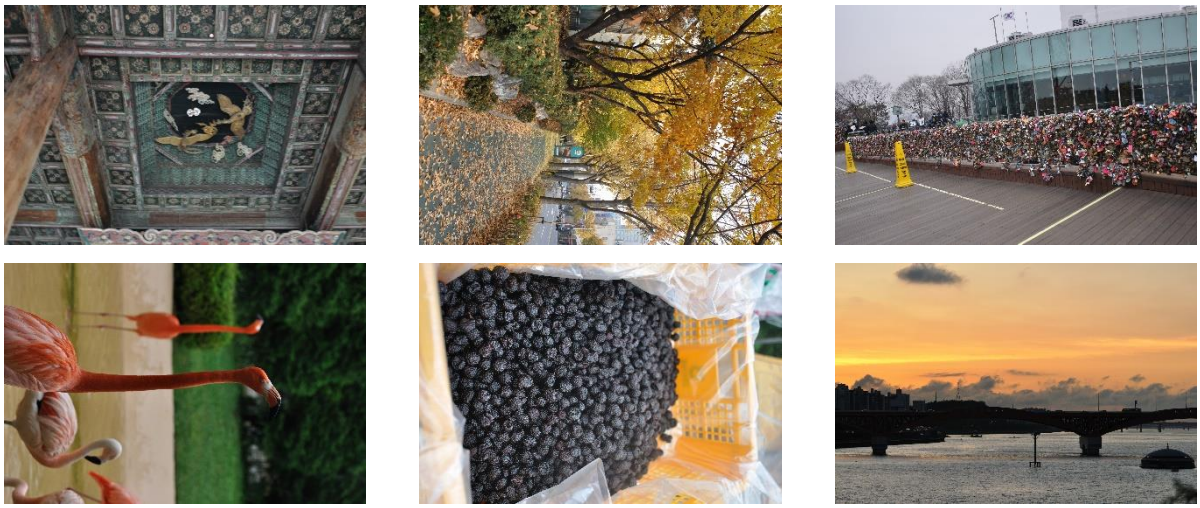


Figure B-2: D90 dataset

Figure B-3: Real CFA dataset

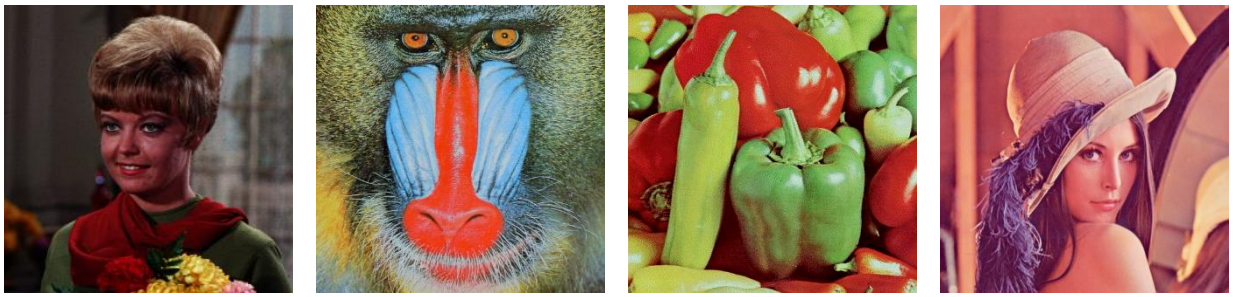


Figure B-4: Standard image dataset

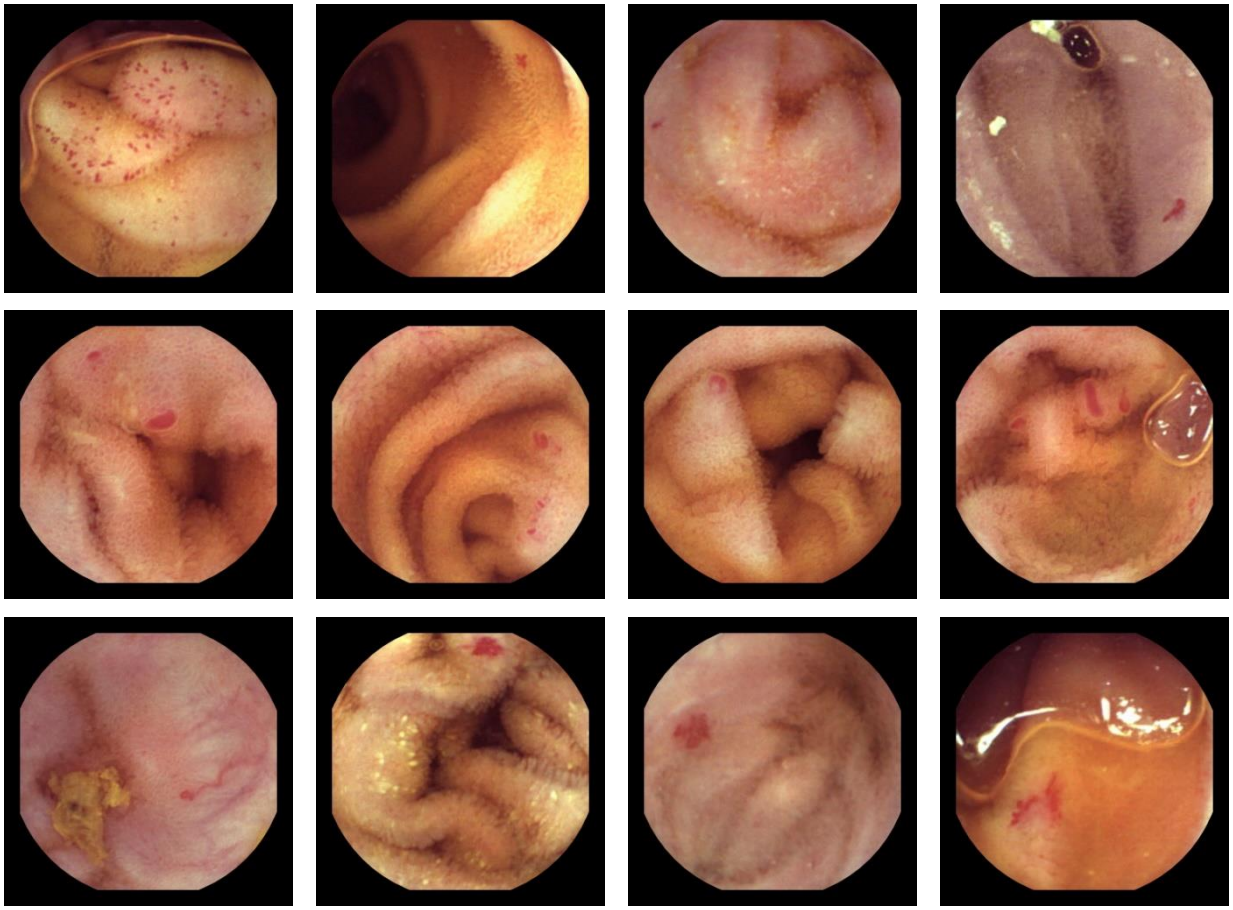


Figure B-5: WCE KID dataset

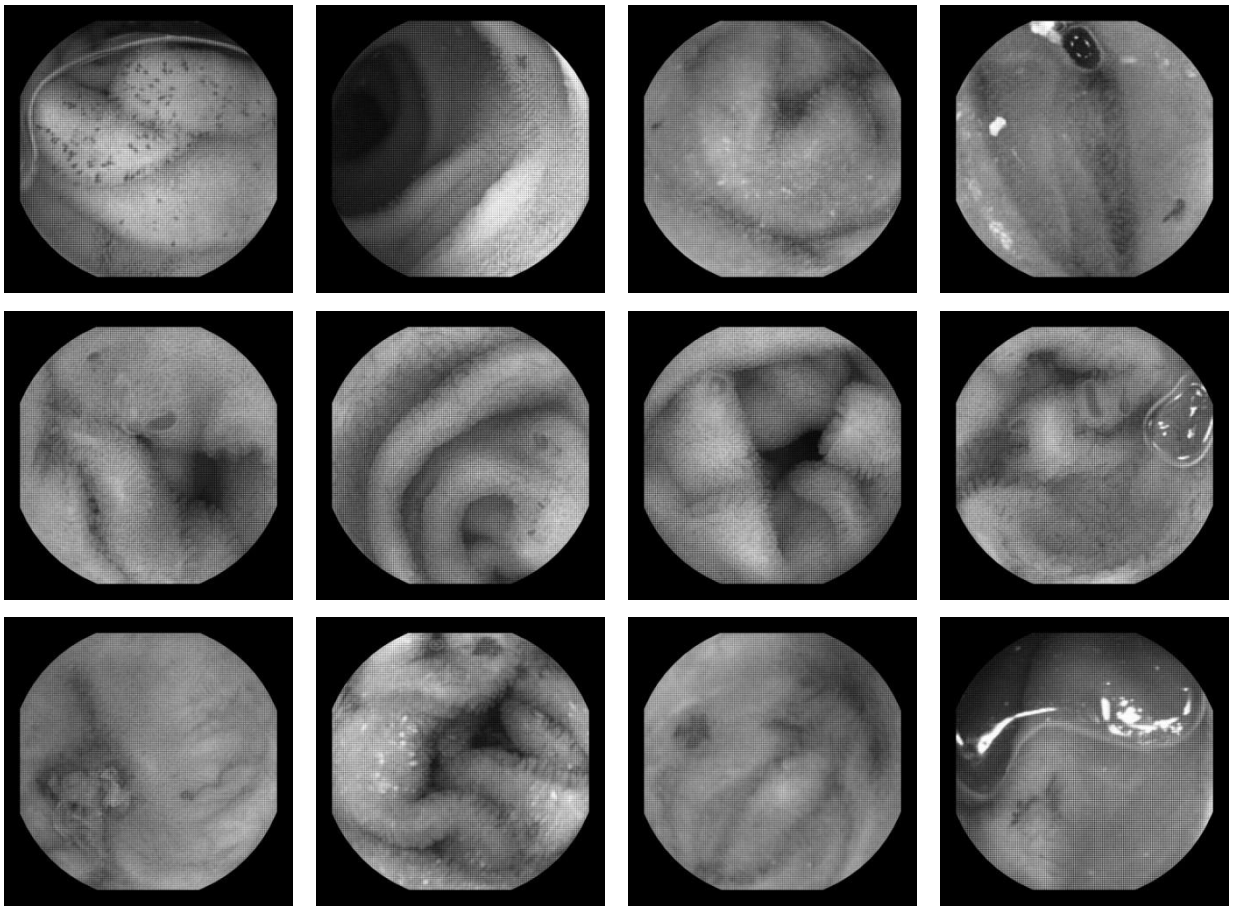


Figure B-6: WCE KID simulated CFA dataset