International Journal of Research in Engineering, Science and Management
Volume-2, Issue-6, June-2019
www.ijresm.com | ISSN (Online): 2581-5792

Color Filter Array Compression using Ranking Method for Lossless Transmission

R. Jayarani
Lecturer, Dept. of Electronics and Communication Engineering, Government Polytechnic College, Trichy, India

***Abstract*: Image compression is the technique which is used to reduce irrelevant and redundant data in the image in order to be able to store or transmit data in an efficient form. This project presents an encoder for the lossless compression of color filter array (CFA) data, which consists of a Discrete Cosine Transform and Arithmetic encoder. In Discrete Cosine Transform, the sub sampled images are encoded in order; each of the sub images contains only one color component (red, green, or blue) in the case of a Bayer CFA image. By sub sampling, the green pixels are separated into two sets, one of which is encoded by a conventional grayscale encoder, and then, it is used to predict the green pixels in the other set. Both the sets of greens are then used to predict the red pixels, and the green and red pixels are used to predict the blues. After that, Discrete Cosine Transform and Arithmetic Encoding algorithm is applied to each and every pixel of an image in transmitter side. In receiver side bit streams are converted, using Arithmetic decoding and Inverse Discrete Cosine Transform. Finally, sub sampled color components are interpolated. This method can achieve low compression ratio with high quality of reconstruction.**

***Keywords*: Color filter array (CFA), lossless compression.**

1. Introduction

Most digital cameras have a single image sensor plane, in which every cell captures a wide wavelength range of light. In order to produce a color image with this sensor, red (R), green (G), or blue (B) filters are regularly placed on the cells, making it a color filter array (CFA) sensor. One of the most widely used color filter patterns is the Bayer pattern shown in Fig. 1. Relative to the cameras that capture R, G, and B at each pixel position using separate sensor planes, CFA cameras require less space for the optical system they employ, use less power, and cost less to produce. However, since only one of the RGB components is available at each cell position in such a CFA-based system, the other two color components at each cell position must be interpolated from neighboring data using demosaicking process. Image compression is minimizing the size in bytes of a graphics file without degrading the quality of the image to an unacceptable level. The reduction in file size allows more images to be stored in a given amount of disk or memory space. It also reduces the time required for images to be sent over the Internet or downloaded from Web pages. CFA image is given as input. CFA is Color Filter Array. Digital Image compression addresses the problem of reducing the amount of data required to represent a digital image. The underlying basis of the reduction process is removal of redundant data.

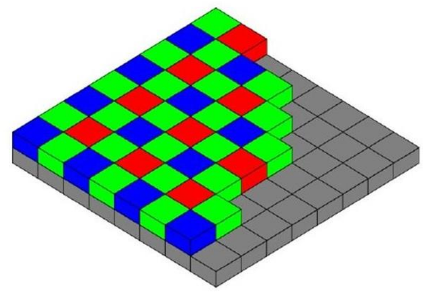

Fig. 1. Bayer Color Filter Mosaic

2. Overview of proposed system

Image compression is minimizing the size in bytes of a graphics file without degrading the quality of the image to an unacceptable level. The reduction in file size allows more images to be stored in a given amount of disk or memory space. It also reduces the time required for images to be sent over the Internet or downloaded from Web pages. CFA image is given as input. CFA is Color Filter Array. Discrete Cosine Transform and Arithmetic encoding are the techniques used to compress the image. Discrete Cosine Transform (DCT) expresses a finite sequence of data points in terms of a sum of cosine functions oscillating at different frequencies. DCT is applied to each and every pixel. Arithmetic coding is a data compression technique that encodes data (the data string) by creating a code string which represents a fractional value on the number line between 0 and 1. The coding algorithm is symbol wise recursive; i.e., it operates upon and encodes (decodes) one data symbol per iteration or recursion. On each recursion the algorithm successively partitions an interval of the number line between 0 and I, and retains one of the partitions as the new interval. Arithmetic coding maps a string of data (source) symbols to a code string in such a way that the original data can be recovered from the code string. The encoding and decoding algorithms perform arithmetic operations on the code string. Interpolation is used to merge the RGB colors. Finally, the output will be a compressed image.

CFA image is given as input. From an input image RGB planes are separated by using plane separation block. After separating the RGB color, apply Direct Cosine Transform to the separated blocks. Encoding is for maintaining data usability and can be reversed by employing the same algorithm that encoded the content. For that Arithmetic encoding is used. After that Resultant pixels are converted to bit streams by using encoder. The bit stream is the compressed format of the image. Then the compressed image is send to the receiver. The arithmetic decoder block decodes the bit stream value into decimal vale. At last the image is reconstructed using Inverse Direct Cosine transform. Interpolation is to merge the separated RGB colors. After interpolation the receiver gets the decompressed image. By comparing input image and reconstructed image the compression Ratio, Mean Square Error, Peak Signal to Noise Ratio values are calculated.

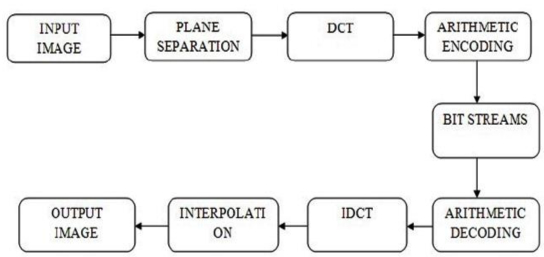

Fig. 2. Structure of proposed system

Fig. 3. Input Image

Fig. 4. Plane separation

Input image is processed by plane separation block. In the input image odd row G pixels are used for the prediction of even row G pixels, both sets of G pixels are used for the prediction of R pixels, and G and R pixels are then used to predict B pixels. After the plane separation apply DCT for every pixel. Color Filter Array is used to separate the single color. Now the input image consists of only a single color (Green).

3. Discrete cosine transform

The Discrete Cosine Transform (DCT) algorithm is well known and commonly used for image compression. DCT converts the pixels in an image, into sets of spatial frequencies. It has been chosen because it is the best approximation of the Karhunen_loeve transform that provides the best compression ratio. The DCT work by separating images into the parts of different frequencies. During a step called Quantization, where parts of compression actually occur, the less important frequencies are discarded, hence the use of the lossy. Then the most important frequencies that remain are used retrieve the image in decomposition process. As a result, reconstructed image is distorted.

A discrete cosine transform (DCT) expresses a finite sequence of data points in terms of a sum of cosine functions oscillating at different frequencies. DCTs are important to numerous applications in science and engineering from lossy compression of audio (e.g. MP3) and images (e.g. JPEG) (where small high-frequency components can be discarded), to spectral methods for the numerical solution of partial differential equations. The use of cosine rather than sine functions is critical for compression, since it turns out that fewer cosine functions are needed to approximate a typical signal, whereas for differential equations the cosines express a particular choice of boundary conditions. In particular, a DCT is a Fourier-related transform similar to the discrete Fourier transform (DFT), but using only real numbers. DCTs are equivalent to DFTs of roughly twice the length, operating on real data with even symmetry (since the Fourier transform of a real and even function is real and even), where in some variants the input and/or output data are shifted by half a sample. Advantages of DCT are implemented in single integrated circuit, ability to pack most information in fewest coefficients, minimizes the block like appearance called blocking artifact that results when boundaries between sub-images become visible.

The forward 2D_DCT transformation is given by the following equation,

$$F(u,v) = \frac{1}{\sqrt{2N}} C_u C_v \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} f(x,y) cos\left[\frac{(2x+1)i\pi}{2N}\right] cos\left[\frac{(2y+1)j\pi}{2N}\right] \quad (1)$$

$$C_u\, C_v = \begin{cases} \frac{1}{\sqrt{2N}} & if\ i,j = 0 \\ 1\ if\ i,j > 0 \end{cases} \quad (2)$$

- f(x,y) is the input image represented in matrix form.
- N is the size of the block.

International Journal of Research in Engineering, Science and Management
Volume-2, Issue-6, June-2019
www.ijresm.com | ISSN (Online): 2581-5792

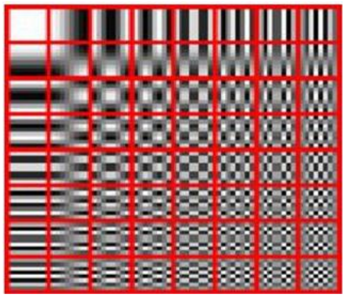

Fig. 5. 8 X 8 Array of Basis Images for the 2d Dct

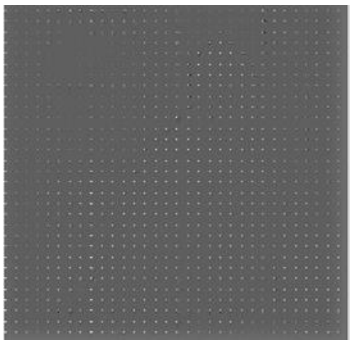

Fig. 6. Dct output

4. Encoding

Encoding is for maintaining data usability and can be reversed by employing the same algorithm that encoded the content, i.e. no key is used. The purpose of encoding is to transform data so that it can be properly (and safely) consumed by a different type of system, e.g. binary data being sent over email, or viewing special characters on a web page.

The goal is not to keep information secret, but rather to ensure that it's able to be properly consumed. Encoding transforms data into another format using a scheme that is publicly available so that it can easily be reversed. It does not require a key as the only thing required to decode it is the algorithm that was used to encode it.

A. Arithmetic encoding

Arithmetic coding is a form of entropy encoding used in lossless data compression. Normally, a string of characters such as the words "hello there" is represented using a fixed number of bits per character, as in the ASCII code. When a string is converted to arithmetic encoding, frequently used characters will be stored with fewer bits and notso-frequently occurring characters will be stored with more bits, resulting in fewer bits used in total. Arithmetic coding differs from other forms of entropy encoding such as Huffman coding in that rather than separating the input into component symbols and replacing each with a code, arithmetic coding encodes the entire message into a single number, a fraction n where ($0.0 \leq n < 1.0$).

In general, arithmetic coders can produce near-optimal output for any given set of symbols and probabilities (the optimal value is −log2P bits for each symbol of probability P, see source coding theorem). Compression algorithms that use arithmetic coding start by determining a model of the data basically a prediction of what patterns will be found in the symbols of the message. The more accurate this prediction is, the closer to optimal the output will be.

B. Calculation for arithmetic encoding

This compression is quite similar to Huffman, because it search for the quite same kind of content to compress, the different are the way it process the source and instead of giving bit value for each symbol it uses probability value for each symbol. It is based on probability between 0(zero) t0 1(one. This method uses a simple math calculation, and this simple calculation resulted in the best compression ratio. It requires five variables for the encoding, they are RANGE, LOW, HIGH, LR (Low Range) and HR (High Range).

1) Encoding

- Define starting variable value: LOW=0; HIGH=1
- Range=HIGH (previous) - LOW(previous)
- LOW=LOW (previous) + (RANGE * LR of current symbol)
- HIGH = LOW (previous) + (RANE * HR of current symbol)

Matrix vale of image pixels are converted into bit streams by arithmetic encoding. Up to arithmetic encoding is sender's part. Sender sends the bit stream value to receiver. After getting bit stream value receiver decodes the data by using arithmetic decoder.

2) Decoding

Require five variables, they are RANGE, LR (Low Range), HR(High Range), VALUE and RD(Range Different).

- Output the symbol by determining which range the VALUE is.
- Get a new VALUE using this calculation
- RD = HR-LR
- VALUE = (VALUE-LR)/RD

C. Inverse discrete cosine transform

The inverse discrete cosine transform (IDCT) decodes an image into the spatial domain from a representation of the data better suited to compaction. IDCT-based decoding forms the basis for current image and video decompression standards. The input to the IDCT comes after the de quantization step and zig-zag positioning. An 8x8 block of input values range from -2048 to 2047 and output values in the range -256 to 255. This information is used to reconstruct the image. Its pixel values range from 0 to 255. The actual formula used depends on whether QP is even or odd, which is specified by the standard to prevent the accumulation of IDCT mismatch errors.

The 2D inverse discrete cosine transform is given by the following formula

$$f(x,y) = \frac{1}{\sqrt{N}} \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} C_u C_v F(u,v) \cos\left[\frac{2x+1(u\pi)}{2N}\right] \cos\left[\frac{2y+1(v\pi)}{2N}\right] \quad (3)$$

F(u,v) is the input for IDCT

International Journal of Research in Engineering, Science and Management
Volume-2, Issue-6, June-2019
www.ijresm.com | ISSN (Online): 2581-5792

N is the size of an image


Fig. 7. Reconstructed Image

Reconstructed image is shown above. The reconstructed image is similar to the input image without any loss. It is quite compact also.

5. Conclusion

We have proposed a new lossless compression algorithm for the Bayer-patterned CFA images. Compression can be improved by using combination of Discrete Cosine Transform and Arithmetic encoding algorithm. Hence, better Peak signal to Noise Ratio can be obtained and Mean Square Error can be reduced. This can be implemented in medical image processing to reduce the storage area up to 20% and lossless compression is achieved. The results show that the proposed method yields less bits per pixel than the transform-based method and other existing prediction based methods.

References

[1] B. K. Gunturk, Y. Altunbasak, and R. M. Mersereau, "Color plane interpolation using alternating projections," IEEE Trans. Image Process., vol. 11, no. 9, pp. 997–1013, Sep. 2002.
[2] B. K. Gunturk, J. W. Glotzbach, Y. Altunbasak, R. W. Schafer, and R. M. Mersereau, "Demosaicking: Color filter array interpolation," IEEE Signal Process. Mag., vol. 22, no. 1, pp. 44–54, Jan. 2005.
[3] B. E. Bayer, "Color imaging array," U.S. Patent 3 971 065, Jul. 1976.
[4] C. C. Koh, J. Mukherjee, and S. K. Mitra, "New efficient methods of image compression in digital cameras with color filter array," IEEE Trans. Consum. Electron., vol. 49, no. 4, pp. 1448–1456, Nov. 2003.
[5] C. Doutre, P. Nasiopoulos, and K. N. Plataniotis, "H.264-based compression of Bayer pattern video sequences," IEEE Trans. Circuits Syst. Video Technol., vol. 18, no. 6, pp. 725–734, Jun. 2008.
[6] D. Lee and K. N. Plataniotis, "Lossless compression of HDR color filter array image for the digital camera pipeline," Signal Processing: Image Communication, vol. 27, no. 6, pp. 637–649, Jul. 2012.
[7] H. S. Malvar and G. J. Sullivan, "Progressive-to-lossless compression of color-filter-array images using macro pixel spectral-spatial transformation," in Proc. DCC, 2012, pp. 3–12.
[8] K. H. Chung and Y. H. Chan, "A lossless compression scheme for Bayer color filter array images," IEEE Trans. Image Process., vol. 17, no. 2, pp. 134–144, Feb. 2008.
[9] K. H. Chung and Y. H. Chan, "A fast reversible compression algorithm for Bayer color filter array images," in Proc. APSIPA, 2009, pp. 825–888.
[10] N. X. Lian, L. Chang, V. Zagorodnov, and Y. P. Tan, "Reversing demosaicking and compression in color filter array image processing: Performance analysis and modeling," IEEE Trans. Image Process., vol. 15, no. 11, pp. 3261–3278, Nov. 2006.
[11] N. Zhang and X. L. Wu, "Lossless compression of color mosaic images," IEEE Trans. Image Process., vol. 15, no. 6, pp. 1379–1388, Jun. 2006.
[12] R. Lukac and K. N. Plataniotis, "Single-sensor camera image compression," IEEE Trans. Consum. Electron., vol. 52, no. 2, pp. 299–307, May 2006.
[13] S. Y. Lee and A. Ortega, "A novel approach of image compression in digital cameras with a Bayer color filter array," in Proc. IEEE Int. Conf. Image Process., Oct. 2001, pp. 482–485.
[14] W.-J. Yang, K.-L. Chung, W.-N. Yang, and L.-C. Lin, "Universal chroma subsampling strategy for compressing mosaic video sequences with arbitrary RGB color filter arrays in H.264/AVC," IEEE Trans. Circuits Syst. Video Technol., vol. 23, no. 4, pp. 591–606, Apr. 2013.