ANALYSING JPEG CODING WITH MASKING

S.-Y. YEO and *Y.-Y. NGUWI

School of Business (IT), James Cook University Singapore

ABSTRACT

The growing trend of online image sharing and downloads today mandate the need for better encoding and decoding scheme. This paper looks into this issue of image coding. Multiple Description Coding is an encoding and decoding scheme that is specially designed in providing more error resilience for data transmission. The main issue of Multiple Description Coding is the lossy transmission channels. This work attempts to address the issue of re-constructing high quality image with the use of just one descriptor rather than the conventional descriptor. This work compare the use of Type I quantizer and Type II quantizer. We propose and compare 4 coders by examining the quality of re-constructed images. The 4 coders are namely JPEG HH (Horizontal Pixel Interleaving with Huffman Coding) model, JPEG HA (Horizontal Pixel Interleaving with Arithmetic Encoding) model, JPEG VH (Vertical Pixel Interleaving with Huffman Encoding) model, and JPEG VA (Vertical Pixel Interleaving with Arithmetic Encoding) model. The findings suggest that the use of horizontal and vertical pixel interleavings do not affect the results much. Whereas the choice of quantizer greatly affect its performance.

KEYWORDS

Jpeg, image, coding, masking, mdc

1. INTRODUCTION

The use of internet and mobile networks increases exponentially. Transporting image and video over the internet while guaranteeing content and speed is a challenging problem. The difficulty can be attributed to the size of image and video are often much larger than other forms of transmission. When an image or video is transmitted through internet, it is first encoded into several packets using progressive encoding scheme like Joint Photographic Expert Group (JPEG). Encoded packets are then delivered using Transmission Control Protocol (TCP), the standard protocol that controls retransmission of lost packets. When a transient error occurs, decoder fail to decode and re-construct the original signal due to loss of packets.

Fast Multiple Description Coding (FMDC) is an encoding and decoding scheme that is specially designed in providing more error resilience for data transmission. It encodes a source into two or more independent bit streams known as descriptor and each descriptor carries crucial information and features of original source. This ensures that receiver is able to recover the original data with certain level of quality even if some descriptions are lost. Quality improves when more descriptions are received and decoded. As these transmission channels are often unreliable, the popularity of MDC coding has increased over the past few years. Multiple Description Coding typically consists of two phases: encoding phase and decoding phase. During encoding phase, the system divides and compresses an input image and develop descriptions to be transmitted through different channels. At decoding phase, if one of the descriptor is received, it will be decoded and original image is re-constructed with a low, but acceptable quality. When more descriptors are received, the image can be better constructed and produce a higher quality image.

The main issue of Multiple Description Coding is the lossy transmission channels. This work attempts to address the issue of re-constructing high quality image with the use of just one descriptor rather than the conventional two descriptors. This paper is organized into several

sections: Section II presents some literatures related to image coding or transforms. Section III presents the system design and results.

2. LITERATURE REVIEW

This section reviews and summarizes some of the common techniques utilized by various researchers trying to improve the current technology. It focuses on two dimensional techniques like Two Dimensional Discrete Cosine Transform (2D DCT), Two Dimensional Discrete Wavelet Transform (2D DWT), Multiple Description Scalar Quantizer (MDSQ), Entropy Encoding and etc.

2.1. Two Dimensional Discrete Cosine Transform (2D DCT)

Discrete Cosine Transform (DCT) [1, 2] was first discovered and developed by Ahmed et. al. [3] in 1974. It is a technique that converts an input signal (e.g. Image's pixel values) into frequency domain. DCT is widely used in image compression like JPEG, which is the first international standard for still image compression established in 1992. After transforming, the important information of an image concentrates at low frequency area. The less important information presents in higher frequency are then discarded. Human eyes are not sensitive to high frequency component. The file size of an image is reduced, however the quality is still acceptable. The disadvantage of 2D DCT is the introduction of blocking artefacts due to discarding of high frequency component during compression [4].

2.2. Two Dimensional Discrete Wavelet Transform (2D DWT)

Due to the blocking effects mentioned in previous section, JPEG introduced another type of image compression standard which is called JPEG2000 that is based on wavelet. Haar wavelet is used. The 2D Haar wavelet works by down sampling input image twice. First time along the rows, then along the columns. The second time using another sub-band. The 2D DWT has gained worldwide recognition in Digital Signal Processing because of the advantages listed below.

- Two dimensional DWT offers better compression rate and better re-construction of image quality compared to 2D DCT.
- It does not have blocking artefacts.
- Rate of compression is scalable with multiple transformation levels.

Although 2D DWT is a better compression technique compared to 2D DCT, still many people choose to go for 2D DCT because the implementation of 2D DWT in terms of hardware and software is far more expensive and complex than 2D DCT [5].

2.3. Pixel Interleaving (PI)

Pixel interleaving (PI) is a technique that divides an image into two or more coarse sub-images and is usually employed before transformation blocks like 2D DCT or 2D DWT. It produce multiple descriptions for transmission of image. At the decoder side, due to the high correlations between adjacent pixels, original image can be re-constructed by interpolation method even if some of the sub-images are lost during transmission. Sub-images can be formed in four ways: horizontal pixel interleaving, vertical pixel interleaving, diagonal pixel interleaving, and dynamic pixel interleaving. Dynamic pixel interleaving produced better result [6], but extra bits are needed. Horizontal and vertical methods are simple, but results are not ideal due to the separation of adjacent pixels of an image. Overall, pixel interleaving is simple to be integrated into other image coding technique, the disadvantage is that if the lost rate of transmission channel is high, the quality of re-constructed image is relatively poor [7].

2.4. Quantization

Quantization is a technique used to reduce the number of bits required to represent an image. It is a lossy compression technique employed in source encoder. Compression of an image can be accomplished by removing redundant information within the image itself, namely spatial redundancy, spectral redundancy, and temporal redundancy. For quantizer, there are scalar quantizer and multiple description scalar quantizer (MDSQ). Scalar quantizer rounds a floating point real value into an integer value depending on the step size. MDSQ generates two coarse descriptions simultaneously with every input image. It aims to re-construct the input image with the highest quality if both descriptions arrived at the decoder side successfully. As shown in Figure 1 is the basic architecture of encoding system using MDSQ.

The performance of MDSQ depends on the spread of index assignment which in turn determines the amount of redundancy that a particular index assignment possesses [8].
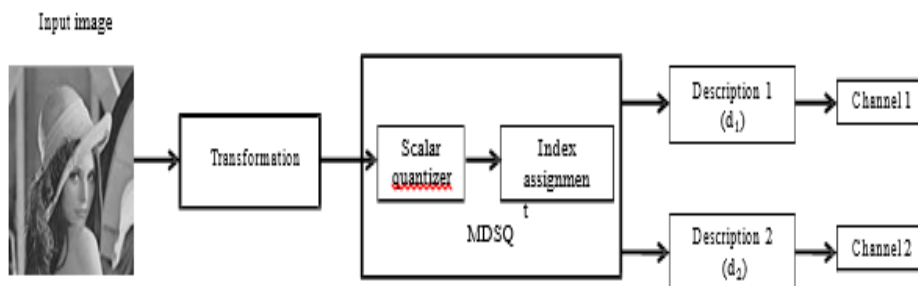

Figure 1. Basic architecture of MDSQ encoding system.

2.5. Entropy Coding

Entropy coding is usually employed quantization to further compress quantized value for better compression ratio. It is a lossless compression scheme that creates and assigns a unique codeword to every unique symbol (pixel value) that occurs within an input image. Entropy coding assigns codeword according to the number of occurrence of a symbol. Symbol that occurs most frequently will be assigned the shortest codeword. Types of entropy coding includes Huffman Coding [9] and Arithmetic Coding [10]. Huffman coding compresses an image by encoding the original data into shorter code, each unique symbol is then given a unique prefix code and length of this code depends on the frequency of occurrence of that particular symbol. These prefix codes are then transmitted for decoding purposes over at the receiver side. Arithmetic coding uses symbol together with probability value that ranges from 0 to 1. The probability denotes the frequency of occurrence.

3. THE SYSTEM AND RESULTS

3.1. JPEG Coding

We assess the performances of the proposed system using compression ratio (CR), mean squared error (MSE) as a measurement of distortion, peak signal to noise ratio (PSNR) and processing time. The higher the compression ratio the image quality would usually be poorer. Higher PSNR measures better quality. A balance of CR and PSNR have to be achieved for optimum image quality. We used three commonly used image processing gray scale, namely cameraman, Lena, and pears. They are of different types, jpg, tif and png. The size and dimensions are listed in Figure 2

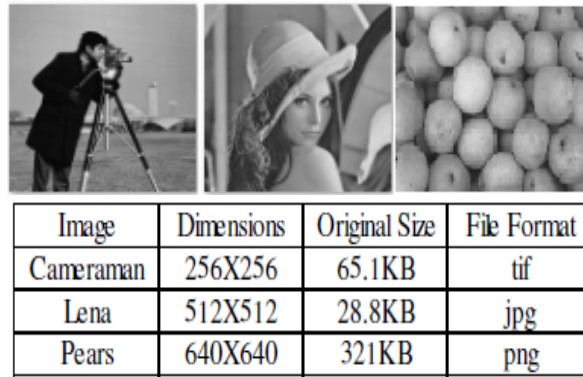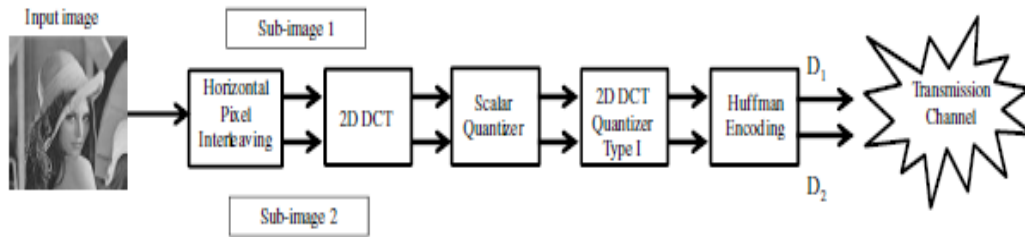| Image	Dimensions	Original Size	File Format
Cameraman	256X256	65.1KB	tif
Lena	512X512	28.8KB	jpg
Pears	640X640	321KB	png

Figure 2. Images used in experiment

Figure 3. JPEG HH encoding model (Horizontal PI and Huffman Encoding)
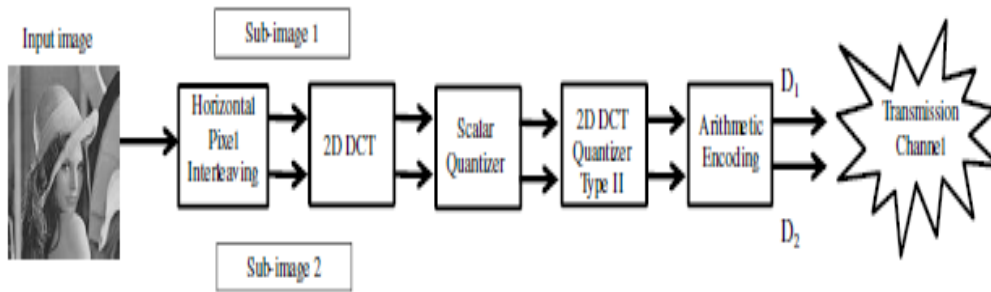
Figure 4. JPEG HA encoding model (Horizontal PI and Arithmetic Encoding)
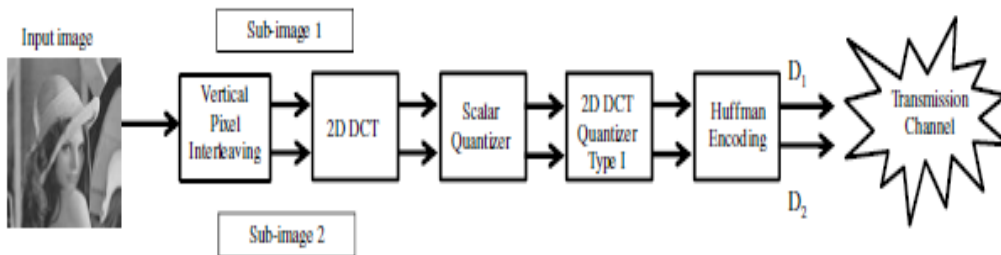
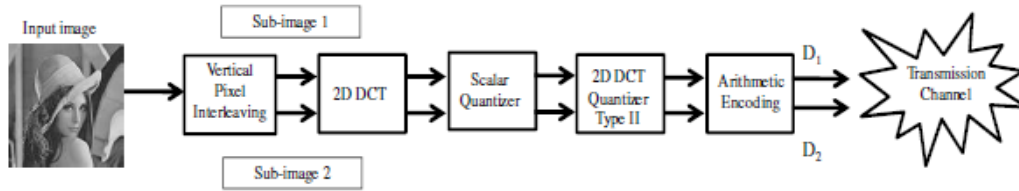Figure 5. JPEG VH encoding model (Vertical PI and Huffman Encoding)

Figure 6. JPEG VA encoding model (Vertical PI and Arithmetic Encoding)
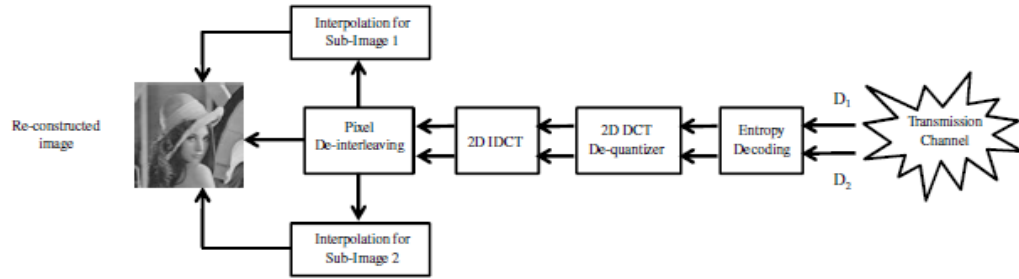

Figure 7. JPEG decoding

Figure 8. Horizontal Pixel Interleaving
. (a) Original Image (512x512) (b) SubImage 1 (256x512) (c) Sub-Image 2 (256x512)

Figure 9. Vertical Pixel Interleaving.
(a) Original Image (512x512) (b) SubImage 1 (512x256) (c) Sub-Image 2 (512x256)

Figure 3 illustrates JPEG HH encoding model using Horizontal Pixel Interleaving and Huffman Encoding. Input image is first interleaved into two sub-images. Figure 4 denotes JPEG HA encoding model using Horizontal Pixel Interleaving and Arithmetic Encoding. Figure 5 JPEG VH model uses Vertical Pixel Interleaving and Huffman Encoding whereas Figure 6 presents the JPEG VA coder using Vertical Pixel Interleaving and Arithmetic Encoding. The sub-images are then fed into 2D DCT for transformation, the transformed images are then quantized into smaller file size. The quantized coefficients are then entropy encoded individually to enhance bandwidth efficiency. Two descriptors D1 and D2 are created for transmission.

Figure 7 illustrates our JPEG decoding system. Decoding reverses the encoding process. Once the system received D1 and D2 descriptors, they will be entropy decoded, de-quantized, and undergone inverse DCT to retrieve back the original image. The interpolation part is needed for the case of only 1 descriptor is received due to transmission error.

The steps for encoding are summarized below:

Step 1: Input image is interleaved into two sub-images.
Step 2: Each sub-image is divided into many 8x8 blocks of pixels.
Step 3: Apply 2D DCT to every block
Step 4: Convert points to integer through scalar quantizer for bytes reduction.
Step 5: Each block is compressed through 2D DCT quantizer to re-presenting original image.
Step 6: After quantization, both sub-images are entropy encoded individually as a whole instead of blocks in generating two descriptions to be transmitted.

As shown in Figure 8 is the original image being interleaved into two sub-images using horizontal pixel interleaving approach. The vertical pixel interleavings are shown in Figure 9. In the next step, the sub-images are then divided into blocks of 8x8 matrixes and each block is transformed into frequency domain for quantization. Figure 10 shows a sample blocks original input and transformed into DCT domain where many of the values are converted into zeros or near zeros. This step has reduced the file size significantly due to this reason. Figure 11 shows the result of using same sub-image without dividing into blocks of 8x8 for 2D DCT. It shows clearly that the block-based method improves the compression ratio by using lesser bytes to represent the same data.

The last step is usually the step where distortion or image information loss are observed. We propose the use of 2 types of quantizer. We propose the use of masking for Type I quantizer. Type II quantizer utilizes JPEG quantizer. Type I masking quantizer takes advantage of the important property of 2D DCT where important information of an image concentrates around lower frequency area. The low frequency components are usually concentrating at the top left area of the matrix, and high frequency components are mostly at the bottom right area of the matrix. We could then improve the compression ratio. We utilized two types of masker, mask8 and mask16. Mask8 keeps the top left 8 coefficients from 2D DCT. Mask16 keeps the top left 16 coefficients from 2D DCT. Each 8x8 block of sub-image is multiplied by either one of them for compression. If a matrix is convolved by mask8, only 8 transformed coefficients remains while the rest are discarded, i.e. only 12.5% (8/64 * 100%). On the other hand, mask16 maintains 25% (16/64 * 100) of the original information. Type II quantizer compresses image by keeping lower frequency components of transformed coefficient. This is done by multiplying the blocks of transformed coefficients with an 8x8 quantization matrix. This matrix allows us to decide on the quality level of the re-constructed image ranging from the level of 1 to 100. Level 1 gives the highest compression but lowest re-constructed quality. On the other hand, level 100 gives the lowest compression but highest quality.

Lastly, Huffman entropy coding was utilized. As shown in Figure 12 are the original matrix and results after inverse transformation. It can be seen that there are high similarity between the 3 matrices in (a), (b) and (c) especially the top left elements of transformed coefficient. This is in-line with the earlier discussion on keeping the lower frequency components in 2D DCT
.

89	89	88	88	88	88	89	89
90	89	89	89	89	89	89	89
90	90	90	90	90	90	90	90
91	91	91	90	90	90	91	92
92	92	92	91	91	91	92	92
93	93	92	92	92	92	92	92
92	92	92	92	92	93	93	93
91	91	91	92	93	93	94	94

Figure 10. JPEG decoding

727	-1	2	0	0	0	0	0
-11	2	0	0	0	0	0	0
-2	-2	-1	1	0	0	0	0
0	2	1	0	0	0	0	0
-1	-1	1	-1	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	-1	0
0	0	0	0	0	0	0	0

Figure 11. Transformed coefficient for block based DCT

Figure 12. Results after inverse transformed with masking
(a) original matrix (b) results from Mask 8 (c) results from Mask 16.

3.2. Performance Comparison

We assess the performances of the proposed system using Compression Ratio (CR), Mean Squared Error (MSE) and Peak Signal to Noise Ratio (PSNR). Table 1 illustrates the results for JPEG decoder using Horizontal Pixel Interleaving and Huffman Encoding with mask8 and mask16. In terms of compression ratio (CR), the performances are within expectation. Mask8 compresses better than mask16 because mask8 uses lesser bits of information. The image type of png has been compressed the most due to its original larger size. In terms of length of codewords, both mask8 and mask16 have about the same size across the three types of images. Whereas for mean square error (MSE), the difference between using 1 descriptor and 2 descriptors do not

differ significantly. The MSE for tif image is much higher than the other 2 types. The other observations to note is the MSE difference between 1 descriptor and 2 descriptors are no more than 10%. This shows that the approach is suitable for even low bandwidth transmission. The signal to noise ratio is better for png image. The use of 1 descriptor and 2 descriptors do not appear to be significant. Our approach achives a good PSNR. The CR can be improved through fine-tuning our quantizer. The experiment also shows that our technique has been getting good results with the use of png files compression. The file size could be shrinked down by 18 times without sacrificing PSNR.

Table 2 shows the results of JPEG HA coder using horizontal pixel interleaving and arithmetic encoding. This coder measures the quality of re-constructed image ranging from level 1 to 100. Level 1 gives the highest compression but lowest re-constructed quality. On the other hand, level 100 gives the lowest compression but highest quality. Hence we measures the bits needed for level 50 and 99. For level 50, the highest compression achieved is 33 times. On the other hand, the best quality level of 99 only obtained 5 times compression for the png image. The MSE for this model appear to be a lot larger than the previous model. The PSNR for all three types of images are lower than the previous model as well.

Table 3 shows the results of JPEG VH coder using vertical pixel interleaving and Huffman encoding. The compression ratio is quite similar to HH. From this table, the MSE for JPG and PNG images are lower and quite similar to JPEG HH decoder. This further suggests that the MSE for Huffman coding is much lower than Arithmetic coding. Table IV shows the results of JPEG VA coder using vertical pixel interleaving and Huffman encoding. With the use of vertical pixel interleaving and Huffman encoding, the results are similar to JPEG HA model, higher MSE and lower PSNR.

The experimental results also show that the use of horizontal pixel interleaving and vertical pixel interleaving give similar results. The difference between the two is less than 1dB for PSNR. Both methods are recommended for 2D DCT system. The other observation is on the choice of quantizers. The compression ratio for type II quantizer (in use for JPEG HA and VA model) is around twice better than type I quantizer (in use for JPEG HH and VH model), the PSNR achieved is around 1.5 times poorer. Type II quantizer compresses an image by dividing every 2D DCT coefficients by a constant. The low frequency area is affected and hence the quality of re-constructed image. Type I quantizer successfully keep lower frequency coefficients without affecting it.

TABLE 1.   JPEG HH CODER RESULTS (Horizontal PI and Huffman Encoding)

Results for JPEG Coder Variant 1	Cameraman.tif (256X256)	Lena.jpg (512X512)	Pears.png (640X640)
CR after Quantization for 1 description	masking8 = 17:1 masking16 = 12:1	masking8 = 2.5:1 masking16 = 1.75:1	masking8 = 18:1 masking16 = 14:1
Length of codeword after Huffman Encoding for 1 description	masking8 = 9197 bits masking16 = 10069 bits	masking8 = 22782 bits masking16 = 22781 bits	masking8 = 24990 bits masking16 = 25007 bits
MSE for 1 description received	masking8 = 336.61 masking16 = 242.42	masking8 = 79.03 masking16 = 44.33	masking8 = 28.01 masking16 = 23.21
MSE for both description received	masking8 = 313.02 masking16 = 200.96	masking8 = 75.92 masking16 = 37.80	masking8 = 26.40 masking16 = 21.32
PSNR for 1 description received	masking8 = 22.86 dB masking16 = 24.29 dB	masking8 = 29.15 dB masking16 = 31.66 dB	masking8 = 33.66 dB masking16 = 34.47 dB
PSNR for both description received	masking8 = 23.18 dB masking16 = 25.10 dB	masking8 = 29.33 dB masking16 = 32.36 dB	masking8 = 33.91 dB masking16 = 34.84 dB

TABLE 2. JPEG HA CODER RESULTS (Horizontal PI and Arithmetic Encoding)

Results for JPEG Coder Variant 2	Cameraman.tif (256X256)	Lena.jpg (512X512)	Pears.png (640X640)
CR after Quantization for 1 description	Level 50 = 13:1 Level 99 = 3:1	Level 50 = 5:1 Level 99 =3:1	Level 50 = 33:1 Level 99 = 5:1
codeword after Arithmetic Encoding for 1 description	Level 50 = 27004 bits Level 99 = 158766 bits	Level 50 = 80479 bits Level 99 = 415855 bits	Level 50 = 140313 bits Level 99 = 897575 bits
MSE for 1 description received	Level 50 = 3136.93 Level 99 = 3112.29	Level 50 = 1358.62 Level 99 = 1351.27	Level 50 = 597.27 Level 99 = 582.85
MSE for both description received	Level 50 = 3107.42 Level 99 = 3060.54	Level 50 = 1353.95 Level 99 = 1343.66	Level 50 = 593.97 Level 99 = 578.50
PSNR for 1 description received	Level 50 = 13.15 dB Level 99 = 13.20 dB	Level 50 = 16.80 dB Level 99 = 16.82 dB	Level 50 = 20.37 dB Level 99 = 20.48 dB
PSNR for both description received	Level 50 = 13.21 dB Level 99 = 13.27 dB	Level 50 = 16.81 dB Level 99 = 16.85 dB	Level 50 = 20.39 dB Level 99 = 20.51 dB

TABLE 3. JPEG VH CODER RESULTS (Vertical PI and Huffman Encoding)

Results for JPEG Coder Variant 3	Cameraman.tif (256X256)	Lena.jpg (512X512)	Pears.png (640X640)
CR after Quantization for 1 description	masking8 = 16:1 masking16 = 12:1	masking8 = 2.6:1 masking16 = 1.6:1	masking8 = 17:1 masking16 = 13:1
Length of codeword after Huffman Encoding for 1 description	masking8 = 9192 bits masking16 = 10064 bits	masking8 = 22779 bits masking16 = 22783 bits	masking8 = 24993 bits masking16 = 25010 bits
MSE for 1 description received	masking8 = 335.59 masking16 = 242.31	masking8 = 79.14 masking16 = 43.80	masking8 = 28.07 masking16 = 23.39
MSE for both description received	masking8 = 313.02 masking16 = 200.96	masking8 = 75.92 masking16 = 37.81	masking8 = 26.38 masking16 = 21.46
PSNR for 1 description received	masking8 = 22.87 dB masking16 = 24.29 dB	masking8 = 29.14 dB masking16 = 31.72 dB	masking8 = 33.65 dB masking16 = 34.44 dB
PSNR for both description received	masking8 = 23.18 dB masking16 = 25.09 dB	masking8 = 29.33 dB masking16 = 32.35 dB	masking8 = 33.94 dB masking16 = 34.81 dB

TABLE 4. JPEG VA CODER (Vertical PI and Arithmetic Encoding)

Results for JPEG Coder Variant 4	Cameraman.tif (256X256)	Lena.jpg (512X512)	Pears.png (640X640)
CR after Quantization for 1 description	Level 50 = 12:1 Level 99 = 3:1	Level 50 = 6:1 Level 99 = 2:1	Level 50 = 33:1 Level 99 = 5:1
Length of codeword after Arithmetic Encoding for 1 description	Level 50 = 27134 bits Level 99 = 158766 bits	Level 50 = 80456 bits Level 99 = 415848 bits	Level 50 = 140313 bits Level 99 = 897575 bits
MSE for 1 description received	Level 50 = 3186.67 Level 99 = 3212.59	Level 50 = 1356.34 Level 99 = 1348.91	Level 50 = 597.27 Level 99 = 582.85
MSE for both description received	Level 50 = 3101.24 Level 99 = 3057.54	Level 50 = 1354.87 Level 99 = 1345.66	Level 50 = 593.97 Level 99 = 578.50
PSNR for 1 description received	Level 50 = 13.06 dB Level 99 = 13.20 dB	Level 50 = 16.80 dB Level 99 = 16.83 dB	Level 50 = 20.37 dB Level 99 = 20.48 dB
PSNR for both description received	Level 50 = 13.22 dB Level 99 = 13.28 dB	Level 50 = 16.81 dB Level 99 = 16.85 dB	Level 50 = 20.39 dB Level 99 = 20.51 dB

4. CONCLUSION

This paper started with introducing Multiple Description Coding and its associated issues. In literature review, we looked at techniques like Two Dimensional Discrete Cosine Transform (2D DCT), Two Dimensional Discrete Wavelet Transform (2D DWT), Multiple Description Scalar Quantizer (MDSQ), Entropy Encoding and etc. We then proposed a system comparing 3 different types of images, jpg, png, and tiff. The system adopted 2D DCT with masking quantization, we used two types of masks: the mask8 and mask16. Mask8 compresses better than mask16 because mask8 uses lesser bits of information. The image type of png has been compressed the most due to its original larger size. In terms of length of codewords, both mask8 and mask16 have about the

same size across the three types of images. Whereas for mean square error (MSE), the difference between using 1 descriptor and 2 descriptors do not differ significantly. The other observations to note is the MSE difference between 1 descriptor and 2 descriptors are no more than 10%. This shows that the approach is suitable for even low bandwidth transmission. The experimental results also show that the use of horizontal pixel interleaving and vertical pixel interleaving give similar results. The difference between the two is less than 1dB for PSNR. Both methods are recommended for 2D DCT system. The other observation is on the choice of quantizers. The compression ratio for type II quantizer (in use for JPEG HA and VA model) is around twice better than type I quantizer (in use for JPEG HH and VH model), the PSNR achieved is around 1.5 times poorer. Type II quantizer compresses an image by dividing every 2D DCT coefficients by a constant. The low frequency area is affected and hence the quality of re-constructed image. Type I quantizer successfully keep lower frequency coefficients without affecting it.

REFERENCES

[1]   Andrew B. Watson,'Image Compression Using the Discrete Cosine Transform', Mathematica Journal Page(s): 81-88. 1994

[2]   Subhasis Saha,'Image Compression – from DCT to Wavelets: A Review', Crossroads, 6(3), March 2000. Pp. 12-21.

[3]   "A brief review of DCT-based JPEG and few novel wavelet-based image coding algorithms in ACM Crossroads", http://www.acm.org/crossroads/xrds6-3/sahaimgcoding.html, access 27 Oct 2014.

[4]   Ahmed, N., Natarajan, T., & Rao, K. R. (1974),'Discrete Cosine Transfom', IEEE Transactions on Computers, C-23(1), 90-93.

[5]   Vaishampayan, V. A. (1993), 'Design of multiple description scalar quantizers', Information Theory, IEEE Transactions on, 39(3), 821-834.

[6]   Anhong, W., Yao, Z., & Jeng-Shyang, P., 'Multiple Description Image Coding Based on DSC and Pixel Interleaving' Paper presented at the Intelligent Information Hiding and Multimedia Signal Processing, 2008. IIHMSP '08 International Conference on. 2008.

[7]   Zixiang, X., Ramchandran, K., Orchard, M. T., & Ya-Qin, Z. (1999), 'A comparative study of DCT- and wavelet-based image coding', Circuits and Systems for Video Technology, IEEE Transactions on, 9(5), 692-695.'

[8]   DeBrunner, V., DeBrunner, L., & Longji, W. (1999)., 'Recovery of lost blocks by dynamic pixel interleaving, Proceedings of the 1999 Paper presented at the Circuits and Systems, 1999. '99 IEEE International Symposium on ISCAS.