International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395 -0056
Volume: 03 Issue: 12 | Dec -2016 www.irjet.net p-ISSN: 2395-0072

Hybrid image compression (lossy+lossless) approach in spatial and DCT domain

Ali Tariq Bhatti[1], Dr. Jung H. Kim[2]
[1]*Department of Electrical & Computer engineering, North Carolina A&T State University, Greensboro, NC, USA*
[2]*Department of Electrical & Computer engineering, North Carolina A&T State University, Greensboro, NC, USA*
Email address: [1]atbhatti@aggies.ncat.edu, [1]alitariq.researcher.engineer@gmail.com, [2]kim@ncat.edu

---

Abstract - *Big Data Technology is categorized in challenges related to large systems and profits. It is always associated with massive amount of data in engineering and other fields in terms of more bandwidth and storage space. This includes the value of 'in-memory' technologies, performing central assessments on compressed data, and increasing performance of utilities such as backups. In order to overcome Big Data Technology, lossy image compression using VQ in DCT domain is implemented where we get better improvement of image data than spatial domain. Different performance metrics have been implemented for the hybrid (combined lossy+lossless (Huffman)) approach of various blocks and codebook size of 25 and 50. In this research paper, a hybrid approach to use vector quantization (VQ) followed by Huffman coding to achieve higher PSNR number in both spatial and Discrete Cosine Transform (DCT) domain. This research is also focused on more improvement in compression ratio using the hybrid approach than the original VQ in spatial and DCT domain.*

Key Words: VQ, K-Means, Huffman, Spatial, DCT.

1.INTRODUCTION

In this research paper, lossy image compression using Vector Quantization (VQ) is used. VQ is that type of quantization in which the blocks of source output are quantized. In addition, error-free reconstruction of the original image may be impossible in many cases for lossy compression. Consequently, lossy compression may produce an acceptable error that does not affect much on the original image. This can be seen in fast transmission of still images over the Internet where the amount of error can be acceptable [1] [2].

The outline of this research paper is that we first discussed the lossy VQ using K-Means in spatial and frequency (DCT) domain. Then, our discussion on the lossless (Huffman coding) approach in spatial and DCT domain. Furthermore, we have the hybrid approach (combined lossy+lossless (Huffman)) in methodology section. At the last, we have results and conclusion section.

2. Lossy VQ for Spatial and DCT domain

Spatial domain are considered more popular as compared to frequency domain as they are involve direct manipulation of pixels in an image and it also enhance the whole image in a uniform manner which can be responsible for any undesirable results [3]. Basically, the idea behind these two domains is to bring out detail that is obscured [4]. In order to improve results, frequency domain (DCT) was used, but one of the problems of the DCT is its blocking effect. In case of DCT, images are divided into blocks of 4x4, 8x8 or larger. The main issue with these blocks is that when the image reduces to high ratios of compression, these blocks become more visible.

2.1 Proposed K-means encoding and decoding process.

K-Means method is numerical, unsupervised, and iterative. In the undergoing research, K-Mean algorithm for block and codebook size in encoding and decoding process always converge as shown in figure 1 and figure 2. A better approach to minimize this problem is to make multiple runs of the algorithm with different K initial seed centroids and choose the best one for a given problem. The two main crucial issues in this research is that K-means clustering model are finding the optimal number of clusters (K) to create, and the initial centroid of each cluster. This research work poses a serious concern which initiated the development of robust strategies for fast convergence of K-means using image compression for block and codebook size for that image. Figure 1 shows the K-Means encoding process as:

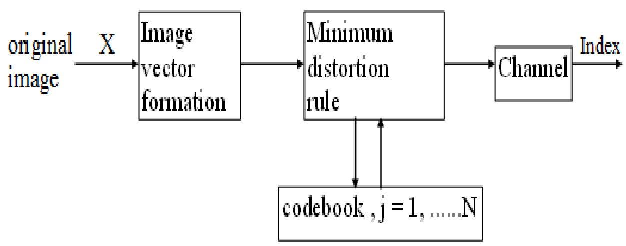

Figure 1. K-Means encoding process [5]

Figure 2 shows the K-Means decoding process as:

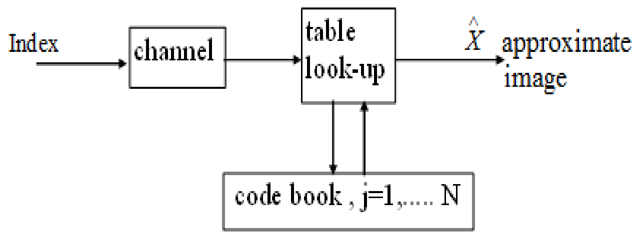

Figure 2. K-Means decoding process [5]

3. Huffman Coding

Huffman coding technique is basically based on frequency of occurrence of an image compression in spatial domain in the undergoing research. It is a classical problem in image processing and computer vision [6] [7]. The principle behind this technique is to use lower number of bits to encode the data that occurs more frequently. A Huffman code dictionary, which associates each data symbol with a code-word, has the property that no code-word is a prefix of any other code-word in the dictionary. The basis for this coding is a code tree according to Huffman, which assigns short code words to symbols frequently used and long code words to symbols. It is also considered as a pre-processing step in many areas like video/image processing applications [8] [9], speech recognition, texture synthesis etc.

The research has proposed to implement the Histogram for Huffman coding in figure 9 and figure 10. Histogram explains the graphical representation of the distribution of numerical data based on continuous variables (quantitative variables). Eventually, the histogram maps luminance, which is defined on how the human eye perceives the brightness of different colors and therefore every pixel in the color or gray-scale image computes to a luminance value between 0 and 255. However, there are major advantages in Huffman coding [10] especially when implemented to process digital images.

3.1. Huffman encoding.

The Huffman encoding starts by constructing a list of all the alphabet symbols in descending order of their probabilities. It then constructs, from the bottom up, a binary tree with a symbol at every leaf. This is done in steps, where at each step two symbols with the smallest probabilities are selected, added to the top of the partial tree, deleted from the list, and replaced with an auxiliary symbol representing the two original symbols [11]. When the list is reduced to just one auxiliary symbol (representing the entire alphabet), the tree is complete. The tree is then traversed to determine the code words of the symbols.

3.2. Huffman decoding.

Huffman algorithm for decoding process is simple and reconstructs the original message or some approximation from the compressed representation. For decoding, the Huffman tree is passed through with the encoded data step by step. It is also possible to write the Huffman tree on the output [12], but this may require more space than just the frequencies. In any case, the decoder must know what is at the start of the compressed file, read it, and construct the Huffman tree for the alphabet. Only then can it read and decode the rest of its input. Start at the root and read the first bit off the input (the compressed file). If it is zero, follow the bottom edge of the tree; if it is one, follow the top edge. Read the next bit and move another edge toward the leaves of the tree. When the decoder arrives at a leaf, it finds there the original, uncompressed symbol, and that code is emitted by the decoder. The process starts again at the root with the next bit.

After the code-words have been created, coding or decoding is accomplished in a simple look up table. The code-word itself is an instantaneous uniquely decodable block code. It is called a block code because each source symbol is mapped into a fixed sequence of code symbols. Whenever a node, not having a successor is reached, the assigned symbol will be written to the decoded data.

3.3. Huffman Coding in spatial and DCT domain

Huffman coding in DCT domain is one of the technique used in this Chapter 3. Huffman coding in DCT deals with block by block basis whereas Huffman coding in spatial domain deals with pixel by pixel basis. As per our results, Huffman coding in DCT domain shows more improvement in image quality and takes less time than Huffman coding in spatial domain.

4. Methodology:

There are two methodologies used in spatial and DCT domain.

4.1. Methodology of hybrid coding using lossy VQ & lossless Huffman in spatial domain.

The steps are based on the combined hybrid approach (lossy+lossless (Hufman) in spatial domain are:

Step 1: Read bridge image of 256x256 and convert it to gray-scale level.

Step 2: Initialization of the size of block 'M' and size of codebook 'N' for different scenarios in spatial domain.

Step 3: Quantizing K-Mean clustering for bridge image in spatial domain

Step 4: Perform Huffman coding after lossy VQ using K-Means clustering in spatial domain for bridge image.

Step5: From the index columns, we compute probability of symbols arranged in descending order and lower probabilities are merged. This step is continued until only two probabilities are left and codes are assigned according to rule that the highest probable symbol will have a shorter length code.

Step 6: Huffman encoding is performed i.e. mapping of the code-words to the corresponding symbols in the codebook table.

Step 7: Apply Huffman decoding process on final encoded values and output Huffman code words.

Step 8: Original image is reconstructed which is compressed and decompression is done by Huffman decoding. The final image did not suffer any degradation because of lossless Huffman procedure in spatial domain. But Compression Ratio changes, as we see the table 2 results.

Step 9: Check the PSNR, SNR, and execution time of one the scenarios for lossy VQ reconstructed image in spatial domain.

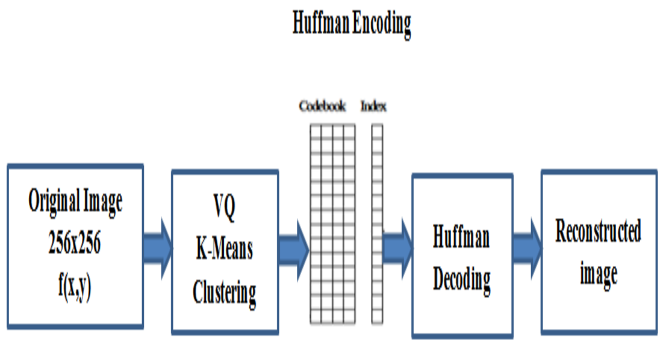

Figure 3. Huffman coding in spatial domain

4.2. Methodology of hybrid coding using lossy VQ & lossless Huffman in DCT domain.

The steps are based on the combined hybrid approach (lossy+lossless (Hufman) in DCT domain are:

Step 1: Read bridge image of 256x256 and convert it to gray-scale level.

Step 2: Initialization of the size of block 'M' and size of codebook 'N' for different scenarios in DCT domain

Step 3: Quantizing K-Mean clustering for bridge image in DCT domain

Step 4: Perform Huffman coding after lossy VQ in DCT domain for both bridge image.

Step 5: Apply DCT on each of the blocks

Step 6: Perform Huffman encoding steps on the blocks using VQ in DCT domain as we implemented in spatial domain (section 4.1 Steps 5 and Step 6).

Step 7: Perform the inverse block DCT (IDCT) and obtain a reconstructed image

Step 8: Original image is reconstructed which is compressed and decompression is done by Huffman decoding. The final image did not suffer any degradation because of lossless Huffman procedure in DCT domain. But Compression Ratio changes, as we see the table 5 results.

Step 9: Check the PSNR, SNR, and execution time of one the scenarios for lossy VQ reconstructed image in DCT domain.

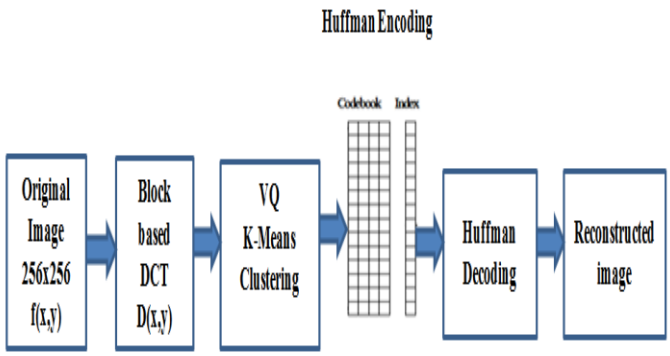

Figure 3. Huffman coding in DCT domain

5. Results:

5.1. Results of K-means using lossy VQ and lossless Huffman in spatial domain.

Table 1 shows the Bridge images of K-Means using lossy VQ in Spatial domain. The detailed information in the table represents different block size used as 4x4 (16), 8x8 (64), 16x16 (256), and 32x32 (1024) with codebook size of 25 and 50. By using these block and codebook sizes, different performance metrics have been calculated to check quality of the Bridge image."

Table 1.
Bridge image of K-Means VQ in Spatial domain

Image 256x256	Bit Rate	Compression Ratio	SNR	PSNR	MSE	Execution Time
1024x25	0.0045	$1.7641\text{x}10^3$	18.0256	21.129	2770.96	6.8045
1024x50	0.0055	$1.4515\text{x}10^3$	18.9541	21.550	2702.45	7.5149
256x25	0.0181	441.0128	19.8573	22.651	2691.32	8.7968
256x50	0.0220	362.8725	20.5265	23.249	2506.66	10.410
64x25	0.0726	110.2532	21.4792	24.952	2476.58	12.0607
64x50	0.0882	90.7181	21.9804	25.480	2419.35	12.5786
16x25	0.2902	27.5633	23.5684	27.236	1401.67	30.9942
16x50	0.3527	22.6795	24.8823	29.029	1395.29	32.2150

Table 2 shows the bridge images of Huffman coding in spatial domain. The detailed information in the table represents different block size used as 4x4 (16), 8x8 (64), 16x16 (256), and 32x32 (1024) with codebook size of 25 and 50. By using these block and codebook sizes, different performance metrics have been calculated to check quality of the bridge image. We also analyze the execution time of each block and codebook size of bridge image in spatial domain.

From table 2, block size of 16 and codebook size of 50 shows less compression ratio, more entropy, higher the average length and Execution time as compared to other scenarios. However, less compression ratio of 1.11 indicates the best quality-acceptable bridge image for one of the best scenario of 16x50.

Table 2
Bridge Image Huffman Coding in Spatial Domain

Image 256x256	Average length	Entropy	Compression Ratio	Execution Time
1024x25	2	1.88	5/2=2.50	47.569
1024x50	2.42	2.30	6/2.42=2.48	49.925
256x25	2.55	2.50	5/2.55=1.96	51.341
256x50	3.17	3.09	6/3.17=1.89	53.535
64x25	3.22	3.15	5/3.22=1.55	54.284
64x50	4.11	3.56	6/4.11=1.46	55.864
16x25	4.42	4.39	5/4.42=1.13	57.726
16x50	5.40	5.38	6/5.40=1.11	59.986

Table 3 shows the overall compression ratio using the hybrid approach (combined lossy and lossless approach) for bridge image in spatial domain.

Table 3.
Overall Compression Ratio for Bridge Image Huffman Coding in Spatial Domain

Image 256x256	Compression Ratio of VQ	Compression Ratio of Huffman	Compression Ratio (overall)
64x25	110.2532	5/3.22=1.55	170.89
64x50	90.7181	6/4.11=1.46	132.45
16x25	27.5633	5/4.42=1.13	31.15
16x50	22.6795	6/5.40=1.11	25.17

From table 1 and table 3, it can be seen that more improvement in VQ design in spatial domain is obtained by using combined lossy and lossless approach. The combined VQ/Huffman approach can achieve performance higher than the original VQ approach in spatial domain. In table 3, overall bridge image compression ratio of 25.17 is higher than the original VQ compression ratio of 22.6795 for block size of 16 and codebook size of 50 in spatial domain.

5.2. Results of K-means using lossy VQ and lossless Huffman in DCT domain.

Table 4 shows the Bridge images of K-Means using VQ in DCT domain. The detailed information in the table represents different block size used as 4x4 (16), 8x8 (64), 16x16 (256), and 32x32 (1024) with codebook size of 25 and 50.

Table 4.
Bridge image of K-Means VQ in DCT domain

Image 256x256	Bit Rate	Compression Ratio	SNR	PSNR	MSE	Execution Time
1024x25	0.0045	$1.7641\text{x}10^3$	19.5554	23.6007	2701.36	6.563
1024x50	0.0055	$1.4515\text{x}10^3$	20.2371	23.9526	2604.47	7.2005
256x25	0.0181	441.0128	21.4876	24.7961	2502.71	8.2518
256x50	0.0220	362.8725	21.8829	25.3305	2391.39	10.0287
64x25	0.0726	110.2532	22.6698	26.658	2232.94	10.4408
64x50	0.0882	90.7181	23.4571	27.644	2150.64	10.5274
16x25	0.2902	27.5633	24.8136	28.312	1264.18	29.319
16x50	0.3527	22.6795	25.6922	30.853	1075.15	30.6587

Table 5 shows the bridge images of Huffman coding in DCT domain. The detailed information in the table represents different block size used as 4x4 (16), 8x8 (64), 16x16 (256), and 32x32 (1024) with codebook size of 25 and 50.

Table 5.
Bridge Image Huffman Coding in DCT Domain

Image 256x256	Average Length	Entropy	Compression Ratio	Execution Time
1024x25	2.23	2.16	2.38	47.227
1024x50	2.52	2.39	2.24	49.501
256x25	2.69	2.61	1.86	51.106
256x50	3.30	3.11	1.82	53.024
64x25	3.44	3.19	1.45	54.157
64x50	4.28	4.15	1.40	55.116
16x25	4.51	4.46	1.10	56.778
16x50	5.76	5.60	1.04	58.751

Table 6 shows the overall compression ratio using the hybrid approach (combined lossy and lossless approach) for bridge image in DCT domain.

Table 6.
Overall Compression Ratio for Bridge Image Huffman Coding in DCT Domain

Image 256x256	Compression Ratio of VQ	Compression Ratio of Huffman	Compression Ratio (overall)
64x25	110.2532	1.45	159.87
64x50	90.7181	1.40	127.01
16x25	27.5633	1.10	30.32
16x50	22.6795	1.04	23.59

From table 4 and table 6, it is evident that more improvement in VQ design in DCT domain are obtained by using combined lossy and lossless approach. The combined VQ/Huffman approach can achieve performance higher than the original VQ approach in DCT domain. In table 6, overall bridge image compression ratio of 23.59 is higher than the original VQ compression ratio of 22.6795 for block size of 16 and codebook size of 50 in DCT domain. The overall compression ratio in spatial domain in table 3 higher than the overall compression ratio in DCT domain in table 6. As by looking the results, table 3 has a higher bridge image overall compression ratio of 25.17 in spatial domain as compare to table 6 bridge image overall compression ratio of 23.59 for block size of 16 and codebook size of 50 in DCT domain.

6. Figures for lossy VQ

Original bridge images are shown in figure 5 as

Figure 5. Original Bridge image

6.1. Figures for lossy VQ in spatial domain.

Figure 6 (a), (b), (c) and (d) represents the reconstructed bridge image for block size of 16 and 64 with codebook size of 25 and 50 in spatial domains.

Figure 6. Bridge image of block 16 and 64 with codebook of 25 and 50 in spatial domain

Figure 7 (a), (b), (c), and (d) represents the reconstructed bridge image for block size of 256 and 1024 with codebook size of 25 and 50 in spatial domains.

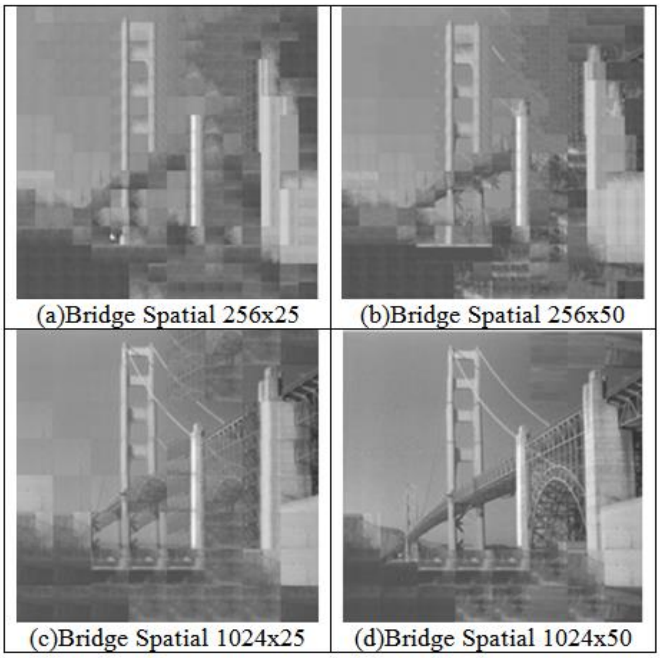

Figure 7. Bridge image of block 256 and 1024 with codebook of 25 and 50 in spatial domain

From these figures 6 (a)-(d) and figures 7 (a)-(d), figure 6 (b) of 16x50 (block size of 16 and codebook size of 50) shows higher SNR, and PSNR to get best quality-acceptable image than other figures in spatial domain. For figure 7 (c) and 7 (d), 1024x25 and 1024x50 reconstructed image gives very higher Compression Ratio, lower SNR, lower PSNR, higher MSE, and less execution time.

6.2. Figures for lossy VQ in DCT domain

Figure 8 (a), (b), (c) and (d) represents the reconstructed bridge image in DCT domain. Figure 8 (a) is the reconstructed bridge image for block size of 16 and codebook size of 50. Figure 8 (b) is the reconstructed bridge image for block size of 64 and codebook size of 50. Figure 8 (c) is the reconstructed bridge image for block size of 256 and codebook size of 50. Figure 8 (d) is the reconstructed bridge image for block size of 1024 and codebook size of 50.

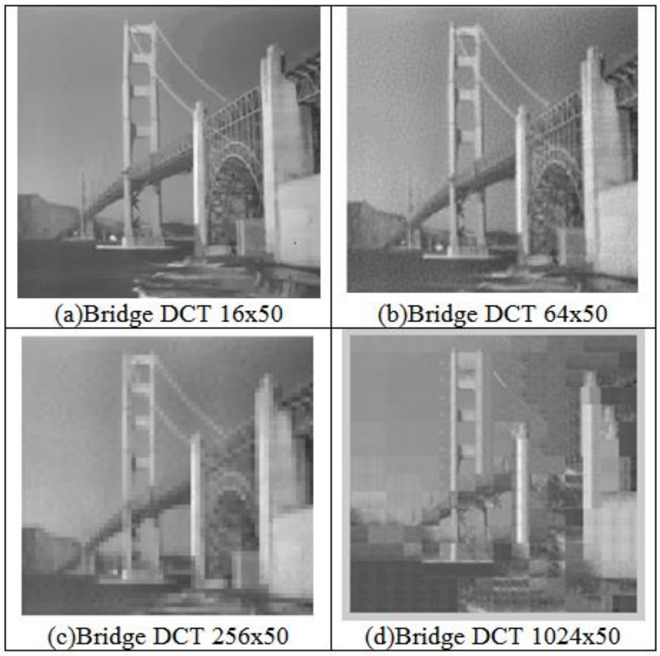

Figure 8. Bridge image of various blocks size with codebook size of 25 and 50 in DCT domain

From these figures 8 (a)-(d) of 16x50 (block size of 16 and codebook size of 50) shows higher SNR, and PSNR to get best quality acceptable image than other figures in DCT domain. Therefore, figure 33 (c) and (d) for block size of 256 and 1024 and codebook size of 25 and 50 shows the worst image quality results for bridge and boat image in DCT domain.

Spatial domain quantizes the image pixel by pixel where DCT domain quantizes the image block by block. In terms of reconstructed boat image, Spatial domain using VQ has a PSNR of 29.913 in table 1 results and DCT domain using VQ has a PSNR of 31.156 in table 4 for block size of 16 and codebook size of 50 which is more than 1 decibel difference between them. To conclude, DCT domain using VQ has a higher PSNR with better boat image quality results than spatial domain using VQ for block size of 16 and codebook size of 50.

6.3 Figures for Histogram of reconstructed Huffman coding 16x50

In figure 9 and figure 10, the horizontal axis is the gray-level values. It begins at zero and goes to the number of gray-scale levels. Each vertical bar represents the number of times the corresponding gray-scale level occurred in the bridge image.

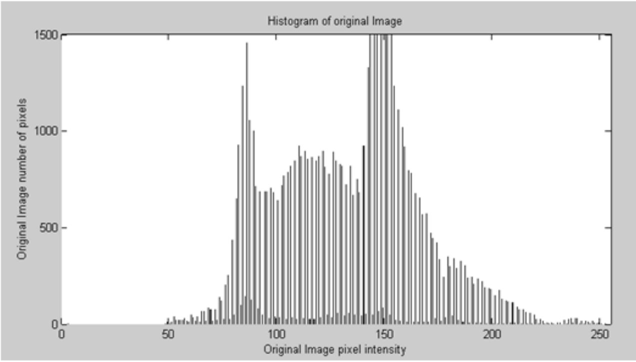

Figure 9. Histogram of original bridge image

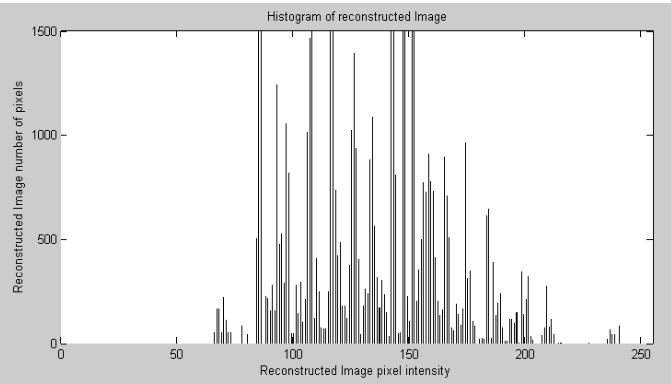

Figure 10. Histogram of reconstructed 16x50 bridge image

Figure 9 and figure 10 are the Histogram for original bridge image and Histogram for reconstructed bridge image of block size of 16 and codebook size of 50.

7. Performance Metrics

There are following performance metrics used for VQ techniques.

7.1 Lossy compression.

(a) Bit rate is defined as:

$$\text{Bit Rate} = \frac{\log_2 N}{M} \quad (1)$$

'M' is the block size and 'N' is the codebook size. The units for bit rate are bits/pixel.

(b) Compression ratio is defined as

$$\text{Compression Ratio} = \frac{\text{Original Bit Rate}}{\text{New Bit Rate}} \quad (2)$$

For scalar in DCT domain:

$$\text{Compression Ratio} = \frac{\text{Number of block size}}{\text{Number of weights used in block size}} \quad (3)$$

(c) MSE. MSE (Mean Square Error) The reconstructed image is ($Y_{i,j}$) and the original image is ($X_{i,j}$). **m** represents the numbers of rows of pixels of the image and i represents the index of that row. **n** represents the number of columns of pixels of the image and j represents the index of that column.

$$\text{MSE} = \frac{\sum_{i=0}^{m-1}\sum_{j=0}^{n-1}[|Y_{(i,j)} - X_{(i,j)}|^2]}{(mxn)} \quad (4)$$

(d) SNR. SNR (Signal-To-Noise Ratio) is defined as:

$$\text{SNR} = \frac{10 * \log_{10} * (\sum_{i=0}^{m-1}\sum_{j=0}^{n-1} x_{(i,j)}^2)}{\sum_{i=0}^{m-1}\sum_{j=0}^{n-1}[Y_{(i,j)} - X_{(i,j)}]^2} \quad (5)$$

(e)PSNR

$$\text{PSNR} = \frac{10 * \log_{10}(256)^2}{\text{MSE}} \quad (6)$$

7.2 Lossless compression (Huffman coding).

(a) Compression Ratio. Compression ratio is defined as:

$$\text{Compression Ratio} = \frac{\text{Original Bit Rate}}{\text{Average Length}} \quad (7)$$

Compression Ratio is Unit-less.

(b) Entropy. For entropy and average length calculation, 'n' is the number of symbols for probabilities used for the codebook size of 25 and 50 in this research. So, entropy 'H' is:

$$\text{Entropy=H= -}\sum_{i=0}^{n-1} p(a_i)\log_2 p_i \quad (8)$$

(c) Average length 'L'. It is the summation of each probability multiplied by number of bits in the codeword. The codeword for each symbol (a_i) is obtained by traversing the binary tree from its root to the leaf corresponding to the symbol.

$$\text{Average Length} = L = \sum_{i=0}^{n-1} p(a_i)n(a_i) \tag{9}$$

8. Conclusion

In this research paper, it is investigated that results of compression ratio for combined lossy and lossless approach in spatial and DCT domain shows more improvement than the original VQ in spatial (table 1) and DCT (table 4) domain . There is less compression ratio in DCT domain than the spatial domain for block size of 16 and codebook size of 50. So, less compression ratio is the best one in DCT domain in this research. Another thing, we also investigated in our results that lossy VQ in DCT domain has higher PSNR, higher SNR and less execution time than the lossy VQ in spatial domain.

The outcomes of this research are:

(a)Simple and lower memory implementation requirement.

(b)Developed to solve in file compression, multimedia, and database applications maintained by google servers.

Future trend is to analyze and to implement image compression in other clustering algorithms and other frequency domains.

REFERENCES

[1] A. Mcandrew (2004), an Introduction to Digital Image Processing With Matlab, Notes for SCM2511 Image Processing I.

[2] H. B. Kekre, T. Saroode & P. Natu (2014), Performance Comparison of Hybrid Wavelet Transform Formed by Combination of Different Base Transforms with DCT on Image Compression, International Journal of Image, Graphics and Signal Processing (IJIGSP) 6 (4) 39-45.

[3] Arun R, Madhu S. Nair, R.Vrinthavani and Rao Tatavarti. "An Alpha Rooting Based Hybrid Technique for Image Enhancement". Online publication in IAENG, 24th August2011.

[4] Rajesh Garg, Bhawna Mittal, Sheetal Garg, "Histogram Equalization Techniques for Image Enhancement", IJECT Vol. 2, Issue 1, March2011, ISSN 2230-9543.

[5] Y. Linde, A. Buzo and R. M. Gray, "An algorithm for vector quantization design," IEEE Trans. Communication, Vol. com-28,pp.84-95, Jan. 1980.

[6] Y.-T. Kim, "Contrast enhancement using brightness preserving bi histogram equalization," IEEE Trans. on Consumer Electronics, vol. 43,no. 1, pp. 1-8, Feb. 1997.

[7] Y. Wang, Q. Chen, and B. Zhang, "Image enhancement based on equal area dualistic sub-image histogram equalization method," IEEE Trans. on Consumer Electronics, vol. 45, no. 1, pp. 68-75, Feb. 1999.

[8] Pei C., Zeng C., and Chang H., "Virtual Restoration of Ancient Chinese Paintings Using Color Contrast Enhancement and Lacuna Texture Synthesis," Computer Journal of IEEE Transactions Image Processing, vol. 13, no. 3, pp. 416-429, 2004.

[9] Pizer M., "The Medical Image Display and Analysis Group at the University of North Carolina: Reminiscences and Philosophy," Computer Journal of IEEE Transactions on Medical Image, vol. 22, no.1, pp. 2-10, 2003.

[10] Abdullah-Al-Wadud, M., Kabir, Md. Hasanul., Dewan, M. Ali Akber. and Chae, Oksam. 2007) "A Dynamic Histogram Equalization for Image Contrast Enhancement" IEEE 2007.

[11] H.B.Kekre, Tanuja K Sarode, Sanjay R Sange (2011) " Image reconstruction using Fast Inverse Halftone & Huffman coding Technique", IJCA, volume 27-No 6, pp.34-40.

[12] Manoj Aggarwal and Ajai Narayan (2000) "Efficient Huffman Decoding", IEEE Trans, pp.936-939.

BIOGRAPHY

Ali Tariq Bhatti had his MSc. in Electrical Engineering from NC A&T State University. He is a PhD candidate. His research interests include Power & System Control, Signal, Image and Video processing, Wireless & Telecommunication, Biosensors, and Software Development.

Dr. Jung H. Kim is a professor in Electrical & Computer Engineering department from NC A&T State University. His research interests include Signal Processing, Image Analysis and Processing, Pattern Recognition, Computer Vision, Digital and Data Communications, Video Transmission and Wireless Communications.