Extensive Analysis of Standard VQ and Multi-Codebook approach using K-Means algorithm in DCT domain

Dr. Ali Tariq Bhatti[1], Dr. Jung H. Kim[2], Dr. Robert Li[3]
[1, 2, 3] Department of Electrical & Computer engineering
[1, 2, 3] North Carolina A&T State University
Greensboro, NC 27411 USA
Email address: [1]alitariq.researcher.engineer@gmail.com, [1]atbhatti@aggies.ncat.edu, [2]kim@ncat.edu, [3]eeli@ncat.edu

Abstract— Big data technologies are one of the vast and hot topics for image compression. In the digital and computing world, information is generated and collected at a rate that rapidly exceeds the boundary range. It is increasing significantly in bio-medical, bio-informatics, engineering and health. Its main goal is to reduce the storage space, cut down the transmission cost and maintain good quality. There are two types of compression used to reduce the data amount which are lossless and lossy compression. For complete parametric study, image vectors or blocks for a codebook size of 25 and 50 were applied in many different scenarios using K-Means algorithm in Discrete Cosine Transform (DCT) domain. The research paper is focused on the Vector Quantization (VQ) techniques on the frequency domain, specifically Discrete Cosine Transform (DCT). So, first the standard VQ method for single codebook is implemented in DCT domain. It has the advantage of packing energy density efficiently. In order to exploit the advantages of working in the DCT domain further, the Multi-Codebook VQ (MCVQ) approach (more than single codebook) was adapted. The actual vector components used from the DCT block was based on certain specified template feature extraction. There are two types MCVQ approaches used such as MCVQ Quadruple approach and MCVQ Dual approach in the DCT domain. MCVQ Quadruple approach is based on one of the four possible categories (Shadow, Horizontal edge, Vertical edge, and Diagonal edge) before it is vector quantized with four separate codebooks. The result revealed to be better than single codebook approach. To gain further efficiency, the researcher tried to reduce the computation burden by using the Dual codebook approach which relies only on two categories (Shadow and Edge). The "class" decisions were also based on the position-based selection methods. Finally, MCVQ Dual approach outperforms the MCVQ Quadruple codebook approach and also the standard VQ method for single codebook in DCT domain in terms of higher PSNR and less execution with the best acceptable quality image.

Index Terms— Compression, K-Means, Vector Quantization, TCVQ, DCT domain, MCVQ Quadruple, MCVQ Dual.

———————————— ◆ ————————————

1 BACKGROUND

Transforming coding studies in DCT domain were implemented on various 2-dimensional speech compression techniques. It is based on compression of signal by removing redundancies present in it. It is a process of transforming signal into compressed or compact form so that the signal could be stored with lesser bandwidth [13].

Past studies on Transformed Vector Quantization (TVQ) were carried out to improve accuracy using variation of Kohonen's self-organization algorithm for vector quantization and manage to save significant training time [11][12]. The main focus was on TVQ using DCT and Vector Quantization with modified Kohonen method for codebook creation. The more interesting TVQ studies were to reduce the bit-rate for data storage or transmission while maintaining an acceptable image quality.

Furthermore, other vector quantization approach in frequency domain performed a vector quantization in DCT domain. In a congested network like the Internet or low bandwidth communication for wireless transmission, image compression at a low bit-rate is necessary. Some Multi-Codebook designs were discussed for certain block sizes, but investigation was for a limited base in terms of block size and codebook size. Evaluation of the compression performance of the Transformed Code Vector Quantization (TCVQ) revealed its superiority over JPEG at low bit-rate [9].

In other past studies, DCT is then applied to each block to convert the spatial domain gray-scale level of pixels into coefficients in frequency domain. After computation of DCT coefficients, their normalization is performed according to a quantization table with different scales provided by

ISSN 2229-5518

the JPEG standard computed by psycho visual evidence [10].

2 INTRODUCTION

However, Big Data is composed of large amounts of data which is different formats. Therefore, high processing speed is necessary [1]. For flexible data analysis, three cases were applied [2]: first, architecture should support several analysis methods, such as statistical analysis, machine learning, data mining and visual analysis. Second, different storage mechanisms should be used because all data cannot be fit in a single type of storage area.

Additionally, the data should be processed differently at various stages. Third, data should be accessed efficiently. Overall, the storage and computing requirements of Big Data analysis demands more advanced techniques to compress images and video, even in the era of cloud computing [3].

Big Data Technology is categorized in challenges related to large systems and profits. Further research was the need of the situation to address these challenges and to improve the display, analysis, and storage of Big Data in an efficient manner.

In order to improve performance for image quality with minimum bandwidth and with less execution time, standard VQ method and MCVQ approaches in DCT domain were used.

The organization/outline structure of the rest of this research paper is as follows. In section 1, the background or literature survey on the standard VQ method and MCVQ approaches in DCT domain. Concept of Big Data Technology for image compression is discussed in section 2. In section 3 and 4, the Vector Quantization (VQ) and K-Means algorithm is illustrated. Mechanism of Standard VQ method and MCVQ Quadruple/Dual approaches used in this research paper are discussed in section 5 and 6. Research experimental results are discussed in section 7. Finally, section 8 contains the Performance metrics and section 9 indicates Conclusion and Future work.

The remaining section is based on Acknowledgement and References.

3 VECTOR QUANTIZATION (VQ)

It is that type of quantization in which the blocks of source output are quantized. It based on principle of block coding and performs better than the Scalar Quantization. It is not widely used because of the difficulty in hardware implementation. It is a fixed to fixed length algorithm. Vector quantization results in a lower distortion than Scalar Quantization for a given rate. When the source output is correlated, vectors of source output values will tend to fall in clusters. A vector quantization is composed of two operations which is the encoder, and the decoder.

Image and video compression are the applications commonly for Vector Quantization.

Encoding and Decoding process of VQ performs in Figure 1 in this way:

1. Each cluster center becomes a codeword in the codebook.
2. Each codeword is assigned an index that represents its location in the codebook.
3. Indexes are used to represent the vectors from the source.
4. The indexes (instead of a much larger codeword) are transmitted to a receiver-side.
5. At the receiver-side, the index is used to retrieve the corresponding codeword from an identical receiver-side codebook.

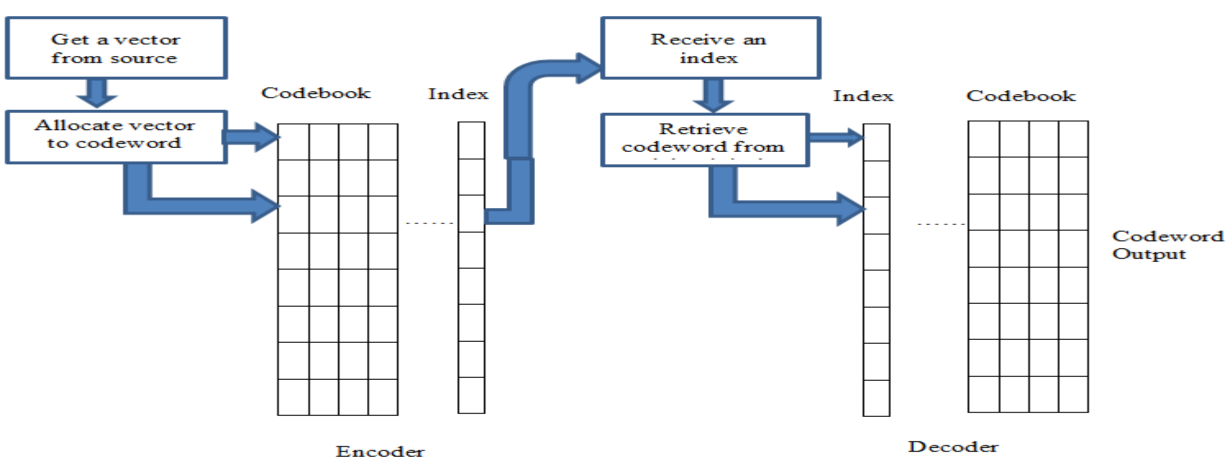

*Figure 1.*Architecture of VQ encoding and decoding process

ISSN 2229-5518

4 K-MEANS Algorithm

K-Means is an unsupervised clustering algorithm used a VQ lossy compression in spatial and frequency domain.

1. Initialization: Randomly K data points are chosen to initialize the cluster centers.
2. Each data point is assigned to the cluster center that is closest to it. It is calculated by the following equation as:

$$b(c_p, a_j) = \sqrt{\sum_{k=1}^{b}(c_{pk} - h_{jk})^2} \quad (1)$$

Where b is the dimension of the data vector, c_p is the centroid of cluster p and h_j is the data vector.

3. New cluster centers are calculated finding the mean of the input vectors assigned to a particular cluster.
4. Stopping rule: repeat steps 2 and 3 until no more change in the value of the means.

5 Standard VQ Approach in DCT domain

Now, lossy image compression using standard VQ method in DCT domain is implemented. The redundant image data is to be removed in lossy compression. High Compression Ratios (CR) are to be achieved and visually negligible difference exists between compressed and original images [4]. The research is to show whether VQ is a more efficient coding technique used in digital image compression area in DCT domain. DCT has the advantage of more efficient energy packing in its frequency domain.

'MxM' 256x256 image is partitioned into many blocks in DCT domain where 'M' is a dimension size in this case. Each sub-image (nxn) is a data vector and its elements are the pixel which sub-image is composed of and where 'n' is the number of blocks. The image is partitioned into many blocks, and each block is considered as a vector [5] as shown in Figure 2. It can provide many attractive features for image coding applications with high CR [5] [6] [7].

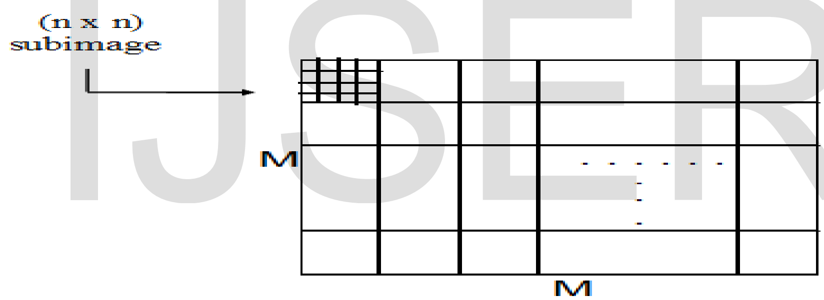

Figure 2. Block partitioning in DCT domain

To build a 2D DCT, use a 1D DCT before over every row and then every column. 1D DCT basically contains an expression of a set of N samples, f(x) as a sum of N cosine basis functions. Therefore, 'x' represents a spatial variable for original image and 'u' represents a frequency variable. 1D Forward DCT (Spatial domain to Frequency domain):

$$F(u) = \frac{2}{N} D(u) \sum_{x=0}^{N-1} f(x) \cos[\frac{\pi u(2x+1)}{2N}] \quad (\mathbf{2})$$

For u=0,1,….,N-1, Where D(k)={$\sqrt{\frac{1}{N}}$ for $u = 0$

$\sqrt{\frac{2}{N}}$ Otherwise}

Therefore, 1D Inverse DCT (Frequency domain to Spatial Domain):

$$f(x) = \frac{2}{N} D(u) \sum_{u=0}^{N-1} F(u) \cos[\frac{\pi u(2x+1)}{2N}] \quad (\mathbf{3})$$

0≤x≤N-1

2D DCT transfer spatial domain to frequency domain. Given an original image in the spatial domain, the pixel coordinates of (x,y) is denoted by f(x,y). The (x,y) represents a 2D spatial coordinates and (u,v) represents the 2D frequency coordinates. To transform 'f' into an image in the frequency domain, 'F', we can use the following:

$$F(u,v) = \frac{2}{N} D(u)D(v) \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} f(x,y) \cos\left[\frac{\pi(2x+1)u}{2N}\right] \cos\left[\frac{\pi(2y+1)v}{2N}\right] \qquad \textbf{(4)}$$

For u=0,1,….,N-1 and v=0,1,…..,N-1

Where $D(k) = \{ \frac{1}{\sqrt{2}}$ for u,v=0
1 Otherwise}

To rebuild an image in the spatial domain from the frequencies obtained above, we use the 2D Inverse DCT (IDCT):

$$f(x,y) = \frac{2}{N} D(u)D(v) \sum_{u=0}^{N-1} \sum_{v=0}^{N-1} F(u,v) \cos\left[\frac{\pi(2y+1)u}{2N}\right] \cos\left[\frac{\pi(2y+1)v}{2N}\right] \qquad \textbf{(5)}$$

for x=0,1,…,N-1 and y=0,1,…,N-1

DCT maps data from the spatial domain to the frequency domain that results in high energy components. Therefore, it would be concentrated in the low-frequency region, so that the transformed vector components in the higher frequency regions have very little information. It shows that the input 'X' was first DCT transformed into Y. Then 'Ŷ' was truncated to get rid of low energy components and vector quantized. The quantized data is inverse transformed to get an approximated form of original 'X̂' [58] as shown in Figure 3.

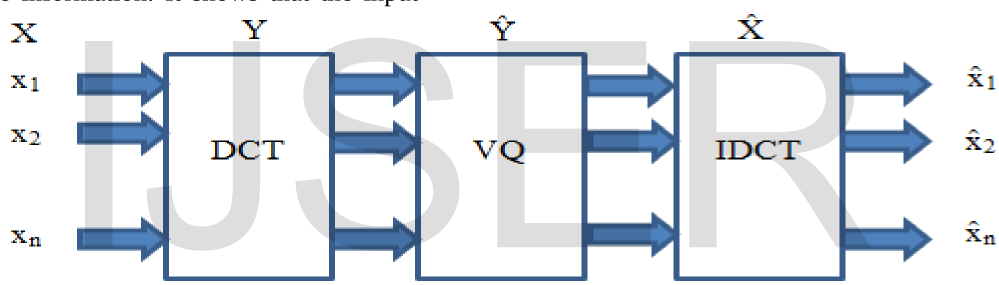

Figure 3. DCT process

5.1 Methodology of Standard VQ in DCT domain.

The following steps summarize the DCT using K-Means VQ algorithm, as shown in Figure 4 and listed in the following steps:

Step 1: In a block by block fashion, the image was transformed to frequency domain using DCT.

Step 2: Perform Vector Quantization (VQ) with K-Means to design the codebook.

Step 3: At receiving end, perform VQ decoding.

Step 4: After decoding the DCT coefficients, the data was inversely transformed using IDCT to obtain a reconstructed image in spatial domain.

Step 5: Implementation of performance metrics such as Bit Rate, CR, SNR, MSE, and PSNR for the reconstructed image in spatial domain based on different scenarios.

Step 6: Check the execution time of all the scenarios for reconstructed image.

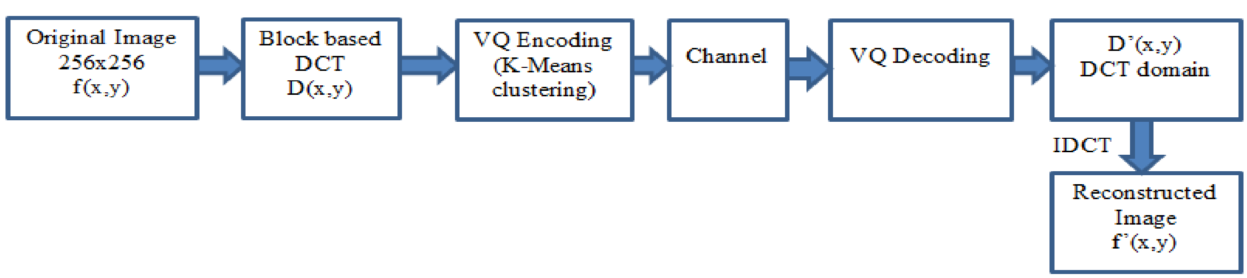

Figure 4. Standard VQ with K-Means clustering in DCT domain

ISSN 2229-5518

6 MCVQ Approach in DCT domain

Multi-Codebook VQ (MCVQ) has the enhanced advantage in DCT domain to maintain the best compressed image quality results with less storage and bandwidth. The advantages of MCVQ technique was used to preserve good reconstructed image quality for edge detection.

The research indicates MCVQ techniques used such as Dual, Quadruple and Adaptive ones in frequency (DCT) domain for different block and codebook size. The research also aimed to propose new procedures for Multi-Codebook Quadruple/Dual approach for background/edge detection.

Researcher has designed the positions for Quadruple codebook approach of different block sizes: 4x4 and 8x8. It classifies whether the block was considered Shadow, Horizontal edge, Vertical edge, or Diagonal edge block. Moreover, the researcher designed the positions for Dual codebook approach of different block sizes: 4x4, 8x8, 16x16, and 32x32. For efficiency, it classifies whether the block is considered shadow or edge block. More than one method is used to make the classification.

VQ lossy compression has always had the high CR and one of the important features is the fast decompression by Table look-up. We want to expand the effort into a Multi-Codebook approach. The purpose of Multi-Codebook is to divide the output of source into different classes with different background/edge properties. In all these classes, it is very significant to design separate vector quantizes for separate classes. Multi-Codebook approach can be considered as a TCVQ approach.

The block diagram of MCVQ is illustrated in Figure 5.

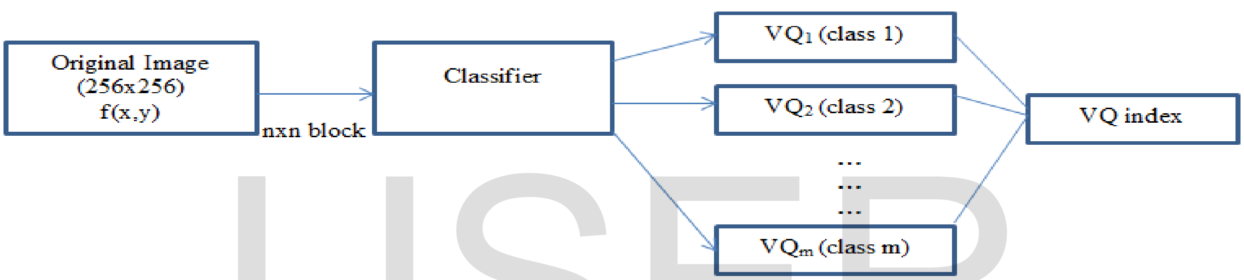

Figure 5. Block diagram of MCVQ

By performing MCVQ in DCT domain, it normally results in a signal energy which is distributed among a small set of transformed coefficients only. We hope to do a better job of classifying pixel blocks into background class and edge class by using these transformed DCT coefficients.

The design of a classifier is based on edge detection in a spatial domain which is not a simple task as it is generally interfered by the background of an image. MCVQ approach in DCT domain takes the advantage of a property of good energy-compaction to simplify the problem of block classification.

MCVQ require much less encoding computation, but also demand fewer bits in transmission and produce better-quality images [8]. However, the scope of its research is limited to only a few parameters of block size and codebook size. This research will do a more in-depth investigation into the task.

6.1 Types of MCVQ Approach

In the undergoing research, there are two types of MCVQ i.e. MCVQ Quadruple approach and MCVQ Dual approach.

6.1.1 MCVQ Quadruple approach. This type of technique has 4 codebooks in DCT domain. It consists of a classifier and separate codebooks for each class. The possible classes in Quadruple approach are typically: Shadow, Horizontal edge, Vertical edge, and Diagonal edge classes.

The design of applying the DCT to different MCVQ Quadruple codebook for any block size like 4x4, 8x8, 16x16, and 32x32 is the following:

- Shadow block produces a DCT matrix with energy concentrated in the top left corner.
- Diagonal edge block produces a DCT matrix with energy concentrated in the diagonal region.
- Horizontal edge block produces a DCT matrix with energy concentrated in the left side.
- Vertical edge block produces a DCT matrix with energy concentrated in the upper side.

Possible edge orientations for Quadruple block type, are the following shown in Figure 6 as:

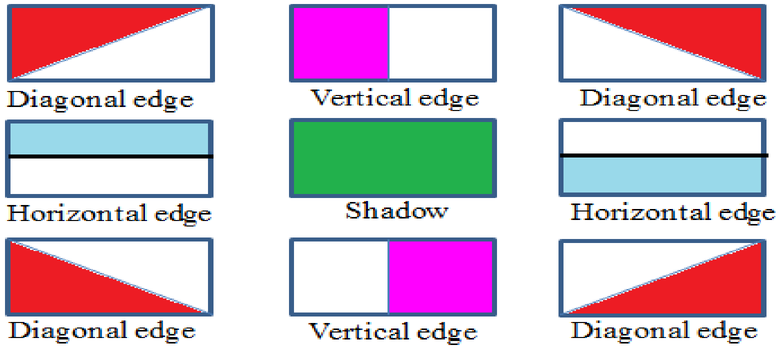

Figure 6. Possible edge orientations for Quadruple block types

6.1.1.1 Design of MCVQ Quadruple Codebooks. One of the most important tasks in a MCVQ Quadruple approach is considered as a design of codebook. All these codebooks are designed in the frequency domain.

For instance, for 4x4: The image is divided into 4x4 blocks and each block is separately transformed using DCT. Then each block has 15 AC coefficients. However, it is too large and more time demanding if we take all the 15 AC coefficients into the code-vector of the training set. Hence, the transformed coefficients will be truncated to eliminate the least important information components. This also reduces the dimension of code-vector in the codebook.

For instance, for 8x8: The image is divided into 8x8 blocks and each block is separately transformed using DCT. Then each block has 63 AC coefficients. However, it is too large and more time demanding if we take all the 63 AC coefficients into the code-vector of the training set. This also reduces the dimension of code-vector in the codebook.

For block of 16 (4x4), the MCVQ Quadruple approach condition setup in Figure 7 is as: H=max($|C_2|$, $|C_3|$), V=max($|C_1|$, $|C_5|$)

C_0	**C_1**	**C_5**	C_6
C_2	C_4	C_7	C_{13}
C_3	C_8	C_{10}	C_{12}
C_9	C_{11}	C_{14}	C_{15}

Figure 7. MCVQ Quadruple conditions setup for block size of 16

Those feature vectors used for different codebooks for block size of 16 are shown in Figure 8 and described below

.

C_0	**C_1**	C_5	C_6
C_2	**C_4**	C_7	C_{13}
C_3	C_8	C_{10}	C_{12}
C_9	C_{11}	C_{14}	C_{15}

(a)Quadruple Shadow 4x4

C_0	C_1	C_5	C_6
C_2	C_4	C_7	C_{13}
C_3	C_8	C_{10}	C_{12}
C_9	C_{11}	C_{14}	C_{15}

(b)Quadruple Horizontal edge block 4x4

C_0	**C_1**	**C_5**	C_6
C_2	**C_4**	C_7	C_{13}
C_3	C_8	C_{10}	C_{12}
C_9	C_{11}	C_{14}	C_{15}

(c)Quadruple Vertical edge block 4x4

C_0	C_1	C_5	C_6
C_2	**C_4**	C_7	C_{13}
C_3	C_8	**C_{10}**	C_{12}
C_9	C_{11}	C_{14}	**C_{15}**

(d)Quadruple Diagonal edge block 4x4

Figure 8. Feature vectors extracted for MCVQ Quadruple codebooks for block size of 16

For 4x4 blocks, the feature vector extracted for the shadow block codebook is (C_1, C_2, C_4). So, a 3-dimensional VQ codebook is used to vector quantize the shadow blocks.

B= Shadow{C_1, C_2, C_4}

Also, the feature vector extracted for the horizontal edge block codebook is (C_2, C_3, C_9). So, a 3-dimensional VQ codebook is used to vector quantize the horizontal edge blocks. B=Horizontal{C_2, C_3, C_9}

Also, the feature vector extracted for the vertical edge block codebook is (C_1, C_4, C_5). So a 3-dimensional VQ codebook is used to vector quantize the vertical edge blocks.

B=Vertical{C_1, C_4, C_5}

Finally, the code-vector for the diagonal edge block codebook is (C_4, C_{10}, C_{15}). So a 3-dimensional VQ codebook is used to vector quantize the diagonal edge blocks.

B=Edge {C_4, C_{10}, C_{15}}

For blocks of 8x8 (64), the MCVQ Quadruple approach condition setup in Figure 9 is as:

H=max(|C_2|,|C_3|,|C_8|,|C_9|), V=max(|C_1|,|C_5|,|C_6|,|C_7|)

C_0	**C_1**	**C_5**	**C_6**	C_{14}	C_{15}	C_{27}	C_{28}
C_2	C_4	**C_7**	C_{13}	C_{16}	C_{26}	C_{29}	C_{42}
C_3	**C_8**	C_{12}	C_{17}	C_{25}	C_{30}	C_{41}	C_{43}
C_9	C_{11}	C_{18}	C_{24}	C_{31}	C_{40}	C_{44}	C_{53}
C_{10}	C_{19}	C_{23}	C_{32}	C_{39}	C_{45}	C_{52}	C_{54}
C_{20}	C_{22}	C_{33}	C_{38}	C_{46}	C_{51}	C_{55}	C_{60}
C_{21}	C_{34}	C_{37}	C_{47}	C_{50}	C_{56}	C_{59}	C_{61}
C_{35}	C_{36}	C_{48}	C_{49}	C_{57}	C_{58}	C_{62}	C_{63}

Figure 9. MCVQ Quadruple conditions setup for block size of 64

Those feature vectors used for different codebooks for block size of 64 are shown in Figure 10 and described below.

C_0	**C_1**	**C_5**	**C_6**	C_{14}	C_{15}	C_{27}	C_{28}
C_2	**C_4**	**C_7**	C_{13}	C_{16}	C_{26}	C_{29}	C_{42}
C_3	**C_8**	C_{12}	C_{17}	C_{25}	C_{30}	C_{41}	C_{43}
C_9	C_{11}	C_{18}	C_{24}	C_{31}	C_{40}	C_{44}	C_{53}
C_{10}	C_{19}	C_{23}	C_{32}	C_{39}	C_{45}	C_{52}	C_{54}
C_{20}	C_{22}	C_{33}	C_{38}	C_{46}	C_{51}	C_{55}	C_{60}
C_{21}	C_{34}	C_{37}	C_{47}	C_{50}	C_{56}	C_{59}	C_{61}
C_{35}	C_{36}	C_{48}	C_{49}	C_{57}	C_{58}	C_{62}	C_{63}

(a)Quadruple Shadow block 8x8

C_0	**C_1**	**C_5**	C_6	C_{14}	C_{15}	C_{27}	C_{28}
C_2	**C_4**	**C_7**	C_{13}	C_{16}	C_{26}	C_{29}	C_{42}
C_3	**C_8**	C_{12}	C_{17}	C_{25}	C_{30}	C_{41}	C_{43}
C_9	**C_{11}**	C_{18}	C_{24}	C_{31}	C_{40}	C_{44}	C_{53}
C_{10}	**C_{19}**	C_{23}	C_{32}	C_{39}	C_{45}	C_{52}	C_{54}
C_{20}	C_{22}	C_{33}	C_{38}	C_{46}	C_{51}	C_{55}	C_{60}
C_{21}	C_{34}	C_{37}	C_{47}	C_{50}	C_{56}	C_{59}	C_{61}
C_{35}	C_{36}	C_{48}	C_{49}	C_{57}	C_{58}	C_{62}	C_{63}

(b)Quadruple Horizontal edge block 8x8

C_0	C_1	C_5	C_6	C_{14}	C_{15}	C_{27}	C_{28}
C_2	C_4	C_7	C_{13}	C_{16}	C_{26}	C_{29}	C_{42}
C_3	C_8	C_{12}	C_{17}	C_{25}	C_{30}	C_{41}	C_{43}
C_9	C_{11}	C_{18}	C_{24}	C_{31}	C_{40}	C_{44}	C_{53}
C_{10}	C_{19}	C_{23}	C_{32}	C_{39}	C_{45}	C_{52}	C_{54}
C_{20}	C_{22}	C_{33}	C_{38}	C_{46}	C_{51}	C_{55}	C_{60}
C_{21}	C_{34}	C_{37}	C_{47}	C_{50}	C_{56}	C_{59}	C_{61}
C_{35}	C_{36}	C_{48}	C_{49}	C_{57}	C_{58}	C_{62}	C_{63}

(c)Quadruple Vertical edge block 8x8

C_0	C_1	C_5	C_6	C_{14}	C_{15}	C_{27}	C_{28}
C_2	C_4	C_7	C_{13}	C_{16}	C_{26}	C_{29}	C_{42}
C_3	C_8	C_{12}	C_{17}	C_{25}	C_{30}	C_{41}	C_{43}
C_9	C_{11}	C_{18}	C_{24}	C_{31}	C_{40}	C_{44}	C_{53}
C_{10}	C_{19}	C_{23}	C_{32}	C_{39}	C_{45}	C_{52}	C_{54}
C_{20}	C_{22}	C_{33}	C_{38}	C_{46}	C_{51}	C_{55}	C_{60}
C_{21}	C_{34}	C_{37}	C_{47}	C_{50}	C_{56}	C_{59}	C_{61}
C_{35}	C_{36}	C_{48}	C_{49}	C_{57}	C_{58}	C_{62}	C_{63}

(d)Quadruple Diagonal edge block 8x8

Figure 10. Feature vectors extracted for MCVQ Quadruple codebooks for block size of 64

For 8x8 blocks, the feature vector extracted for the shadow block codebook is (C_1, C_2, C_3, C_4, C_5, C_6, C_7, C_8, C_9). So, a 9-dimensional VQ codebook is used to vector quantize the shadow blocks.

B=Shadow {C_1, C_2, C_3, C_4, C_5, C_6, C_7, C_8, C_9}

Also, the feature vector extracted for the horizontal edge block codebook is (C_1, C_2, C_3, C_4, C_5, C_7, C_8, C_9, C_{10}, C_{11}, C_{19}). So, a 11-dimensional VQ codebook is used to vector quantize the horizontal edge blocks.

B=Horizontal {C_1, C_2, C_3, C_4, C_5, C_7, C_8, C_9, C_{10}, C_{11}, C_{19}}

Also, the feature vector extracted for the vertical edge block codebook is (C_1,C_2, C_3, C_4, C_5, C_6, C_7, C_8, C_{13}, C_{14}, C_{16}). So, a 11-dimensional VQ codebook is used to vector quantize the vertical edge blocks.

B=Vertical {C_1,C_2, C_3, C_4, C5, C_6, C_7, C_8, C_{13}, C_{14}, C_{16}}

Finally, the code-vector for the diagonal edge block codebook is (C_1,C_2, C_3, C_4, C_5, C_7, C_8, C_{11}, C_{12}, C_{13}, C_{17}, C_{18}, C_{203}, C_{24}, C_{25}}. So, a 15-dimensional VQ codebook is used to vector quantize the diagonal edge blocks.

B=Edge {C_1, C_2, C_3, C_4, C_5, C_7, C_8, C_{11}, C_{12}, C_{13}, C_{17}, C_{18}, C_{23}, C_{24}, C_{25}}

6.1.1.2 Methodology of MCVQ Quadruple approach. MCVQ Quadruple approach classifies the DCT coefficients into one of these four classes which include Shadow, Diagonal edge, Horizontal edge, and Vertical edge. For horizontal edge block, the energy remains concentrated in a left region of the transformed matrix. For vertical edge block, the energy remains concentrated in an upper region of the transformed matrix. In the case of a diagonal edge block, the energy remains concentrated in the diagonal region of transformed matrix. If all the AC upper left coefficients are small, the block is considered as a shadow block.

The following steps summarize the methodology of MCVQ Quadruple approach and are shown in Figure 11.

Step 1: Divide the 256x256 image into one of the block sizes of either 4x4, 8x8, or bigger. Set the codebook size of MCVQ Quadruple in an equally distributed form.

(i) For codebook size of 25: set all four types of codebooks to a size of 6

(ii) For codebook size of 50: set all four types of codebooks to a size of 12,

Step 2: Apply DCT to each block for MCVQ Quadruple approach.

Step 3: MCVQ Quadruple Initialization and Classification steps:

Create four codebooks for block types of Shadow, Diagonal edge, Horizontal edge and Vertical edge.

(i) Compute V=max(|C_1|,|C_5|) and H=max(|C_2|,|C_3|) for block 4x4. Or Compute V=max(|C_1|,|C_5|,|C_6|,|C_7|) and H=max(|C_2|,|C_3|,|C_8|,|C_9|) for block 8x8 or other block size.

'V' and 'H' are the maximum value for the absolute DCT coefficient values.

(ii) Classify the block 'B' into one of the four classes:

If (V<Threshold) and (H<Threshold); so B=Shadow block

Else if (V>=Threshold) and (H>=Threshold) and (max(V,H)/min(V,H)<2); so B=Diagonal edge block

Else if (H>=V); so B=Horizontal edge block

Else B=Vertical edge block

Where **Threshold (T)** = Absolute Average of that block either 4x4, 8x8, 16x16, 32x32

Step 4: At receiving end, perform VQ decoding for each of the codebook and combine together to make a compressed DCT image.
Step 5: After decoding the DCT coefficients, the data was inversely transformed using IDCT to obtain a reconstructed image in the spatial domain.
Step 6: Implementation of performance metrics for the reconstructed image in spatial domain based on different scenarios.

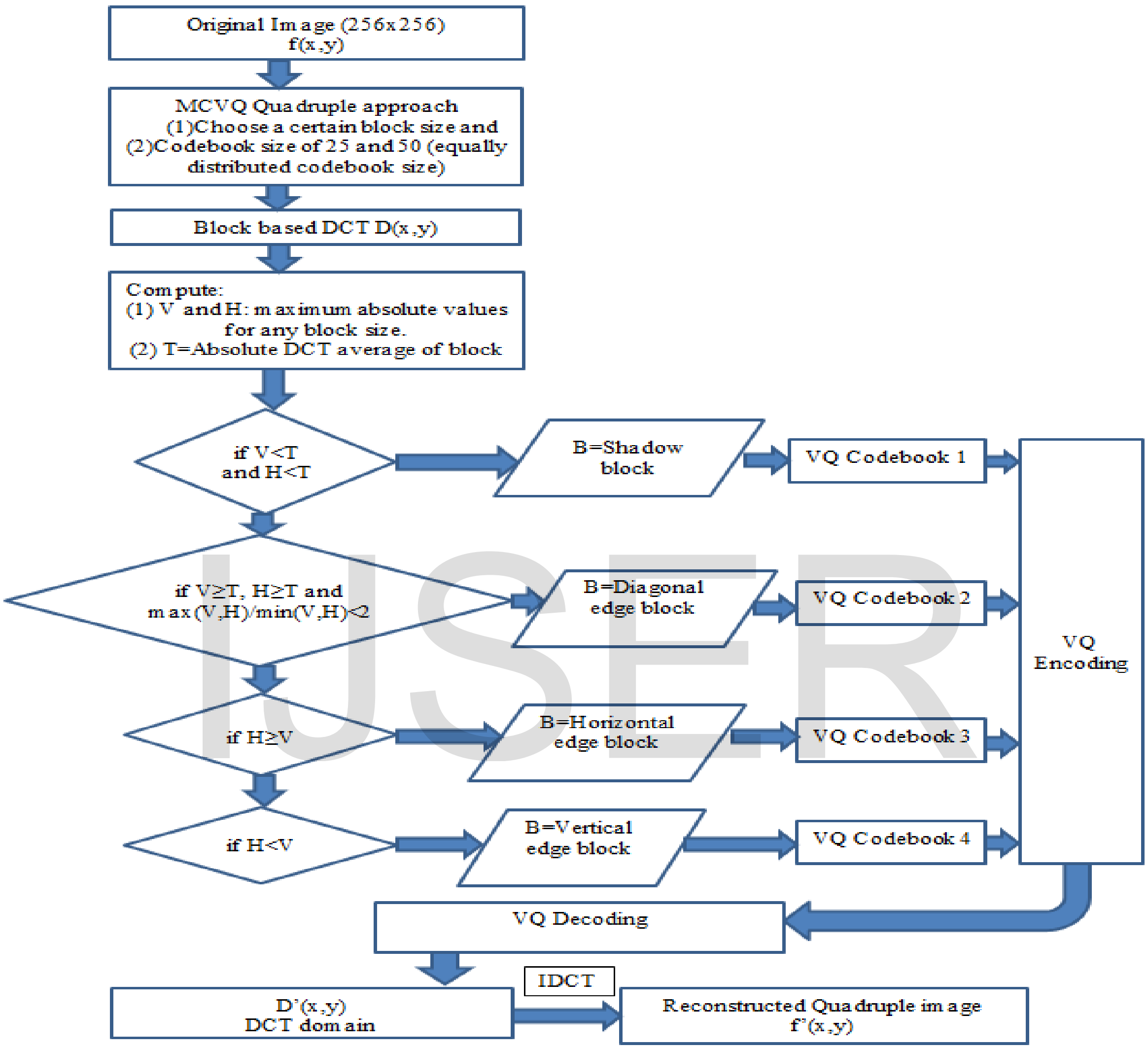

Figure 11. MCVQ Quadruple approach using K-Means in DCT domain

6.1.2 MCVQ Dual approach. MCVQ Dual approach contains 2 codebooks in DCT domain. It consists of a classifier and separate codebooks for each class. The possible classes in Dual approach are typically: Shadow and Edge classes.

The design of applying the DCT to different MCVQ Dual codebook for any block size like 4x4, 8x8, 16x16, and 32x32 is the following. We are producing two edge block types such as Shadow and Edge.

Possible edge orientations for Dual block type, are the following shown in Figure 12 as:

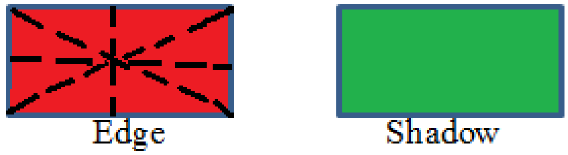

Figure 12. Possible edge orientations for Dual block types

6.1.2.1 Design of MCVQ Dual Codebooks.
One of the most important tasks in a MCVQ Dual approach is the design of the codebook in the frequency domain.

For blocks of 4x4 (16), the MCVQ Dual approach condition setup in Figure 13 is as:
H=max($|C_2|$, $|C_3|$, $|C_9|$), V=max($|C_1|$, $|C_5|$, $|C_6|$)

C_0	C_1	C_5	C_6
C_2	C_4	C_7	C_{13}
C_3	C_8	C_{10}	C_{12}
C_9	C_{11}	C_{14}	C_{15}

Figure 13. MCVQ Dual approach condition setup for block size of 16

MCVQ Dual approach with 2 codebooks for block size of 16 in Figure 14 is as:

C_0	C_1	C_5	C_6
C_2	C_4	C_7	C_{13}
C_3	C_8	C_{10}	C_{12}
C_9	C_{11}	C_{14}	C_{15}

(a)Dual Shadow block 4x4

C_0	C_1	C_5	C_6
C_2	C_4	C_7	C_{13}
C_3	C_8	C_{10}	C_{12}
C_9	C_{11}	C_{14}	C_{15}

(b)Dual Edge block 4x4

Figure 14. Feature vectors extracted for MCVQ Dual codebooks for block size of 16

For 4x4 blocks, the feature vector extracted for the shadow block codebook is (C_1, C_2, C_4, C_5, C_7}. So, a 5-dimensional VQ codebook is used to vector quantize the shadow blocks.
B=Shadow{C_1, C_2, C_4, C_5, C_7}

Finally, the code-vector for the edge block codebook is {C_2, C_4, C_7, C_{10}, C_{15}}. So, a 5-dimensional VQ codebook is used to vector quantize the edge blocks.
B=Edge{ C_2, C_4, C_7, C_{10}, C_{15}}

For blocks of 8x8 (64), the MCVQ Dual approach condition setup in Figure 15 is as:
H=max($|C_2|$, $|C_3|$, $|C_8|$, $|C_9|$, $|C_{10}|$, $|C_{11}|$), V=max($|C_1|$, $|C_5|$, $|C_6|$, $|C_7|$, $|C_{13}|$, $|C_{14}|$)

C_0	C_1	C_5	C_6	C_{14}	C_{15}	C_{27}	C_{28}
C_2	C_4	C_7	C_{13}	C_{16}	C_{26}	C_{29}	C_{42}
C_3	C_8	C_{12}	C_{17}	C_{25}	C_{30}	C_{41}	C_{43}
C_9	C_{11}	C_{18}	C_{24}	C_{31}	C_{40}	C_{44}	C_{53}
C_{10}	C_{19}	C_{23}	C_{32}	C_{39}	C_{45}	C_{52}	C_{54}
C_{20}	C_{22}	C_{33}	C_{38}	C_{46}	C_{51}	C_{55}	C_{60}
C_{21}	C_{34}	C_{37}	C_{47}	C_{50}	C_{56}	C_{59}	C_{61}
C_{35}	C_{36}	C_{48}	C_{49}	C_{57}	C_{58}	C_{62}	C_{63}

Figure 15. MCVQ Dual conditions setup for block size of 64

Those feature vectors used for different codebooks for block size of 64 are shown in Figure 16 and described below.

C_0	**C_1**	**C_5**	**C_6**	**C_{14}**	C_{15}	C_{27}	C_{28}
C_2	**C_4**	**C_7**	**C_{13}**	C_{16}	C_{26}	C_{29}	C_{42}
C_3	**C_8**	**C_{12}**	C_{17}	C_{25}	C_{30}	C_{41}	C_{43}
C_9	**C_{11}**	**C_{18}**	C_{24}	C_{31}	C_{40}	C_{44}	C_{53}
C_{10}	**C_{19}**	**C_{23}**	C_{32}	C_{39}	C_{45}	C_{52}	C_{54}
C_{20}	C_{22}	C_{33}	C_{38}	C_{46}	C_{51}	C_{55}	C_{60}
C_{21}	C_{34}	C_{37}	C_{47}	C_{50}	C_{56}	C_{59}	C_{61}
C_{35}	C_{36}	C_{48}	C_{49}	C_{57}	C_{58}	C_{62}	C_{63}

(a)Dual Shadow block 8x8

C_0	**C_1**	**C_5**	C_6	C_{14}	C_{15}	C_{27}	C_{28}
C_2	**C_4**	**C_7**	**C_{13}**	C_{16}	C_{26}	C_{29}	C_{42}
C_3	**C_8**	**C_{12}**	**C_{17}**	**C_{25}**	C_{30}	C_{41}	C_{43}
C_9	C_{11}	**C_{18}**	**C_{24}**	**C_{31}**	C_{40}	C_{44}	C_{53}
C_{10}	C_{19}	C_{23}	**C_{32}**	**C_{39}**	**C_{45}**	C_{52}	C_{54}
C_{20}	C_{22}	C_{33}	C_{38}	C_{46}	**C_{51}**	**C_{55}**	C_{60}
C_{21}	C_{34}	C_{37}	C_{47}	C_{50}	C_{56}	**C_{59}**	C_{61}
C_{35}	C_{36}	C_{48}	C_{49}	C_{57}	C_{58}	C_{62}	**C_{63}**

(b)Dual Edge block 8x8

Figure 16. Feature vectors extracted for MCVQ Dual codebooks for block size of 64

For 8x8 blocks, the feature vector extracted for the shadow block codebook is $\{C_1,C_2,C_3,C_4,C_5,C_6,C_7,C_8,C_9,C_{10},C_{11},C_{12},C_{13},C_{14},C_{18},C_{19},C_{23}\}$. So, a 17-dimensional VQ codebook is used to vector quantize the shadow blocks.

$B=Shadow\{C_1,C_2,C_3,C_4,C_5,C_6,C_7,C_8,C_9,C_{10},C_{11},C_{12},C_{13},C_{14},C_{18},C_{19},C_{23}\}$

Finally, the code-vector for the edge block codebook is $\{C_1,C_2,C_3,C_4,C_5,C_7,C_8,C_{12},C_{13},C_{17},C_{18},C_{24},C_{25},C_{31},C_{32},C_{39},C_{45},C_{51},C_{55},C_{59},C_{63}\}$. So, a 21-dimensional VQ codebook is used to vector quantize the edge blocks.

$B=Edge\{C_1,C_2,C_3,C_4,C_5,C_7,C_8,C_{12},C_{13},C_{17},C_{18},C_{24},C_{25},C_{31},C_{32},C_{39},C_{45},C_{51},C_{55},C_{59},C_{63}\}$

6.1.2.2 Methodology of MCVQ Dual approach. MCVQ Dual approach classifies the DC coefficients into one of the two classes; Shadow and Edge. If all of the AC upper left coefficients are relatively small, the block must be a shadow block. Otherwise, the block is considered as an edge block.

The following steps summarize the methodology of MCVQ Dual approach and are shown in Figure 17.

Step 1: Divide the 256x256 image into one of the block sizes of either 4x4, 8x8, or bigger. Set the codebook size of MCVQ Dual approach in an equally distributed form.

(i) For codebook size of 25: set both codebooks to a size of 12.

(ii) For codebook size of 50: set both codebooks to a size of 25.

Step 2: Apply DCT to each block for MCVQ Dual approach.

Step 3: MCVQ Dual Initialization and Classification steps

Use DCT block to decide either a Shadow or an Edge block for MCVQ Dual approach.

(i) Compute $V=max(|C_1|,|C_5|,|C_6|)$ and $H=max(|C_2|,|C_3|,|C_9|)$ for block 4x4 or

Compute $V=max(|C_1|,|C_5|,|C_6|,|C_7|,|C_{13}|,|C_{14}|)$ and $H=max(|C_2|,|C_3|,|C_8|,|C_9|,|C_{10}|,|C_{11}|)$ for block 8x8 or other block size. 'V' and 'H' are the maximum value for the absolute DCT coefficient values.

(ii) Classify the block B into one of the two classes

If (V<Threshold) and (H<Threshold); so, B=Shadow block

Else B=Edge block

Where **Threshold (T)** = Absolute Average of the block of either 4x4, 8x8, 16x16, 32x32

Step 4: At receiving end, perform VQ decoding for each of the codebook and combine together to make a compressed DCT image.

Step 5: After decoding the DCT coefficients, the data was inversely transformed using IDCT to obtain a reconstructed MCVQ Dual image in the spatial domain.

Step 6: Implementation of performance metrics for the reconstructed image in spatial domain based on different scenarios.

ISSN 2229-5518

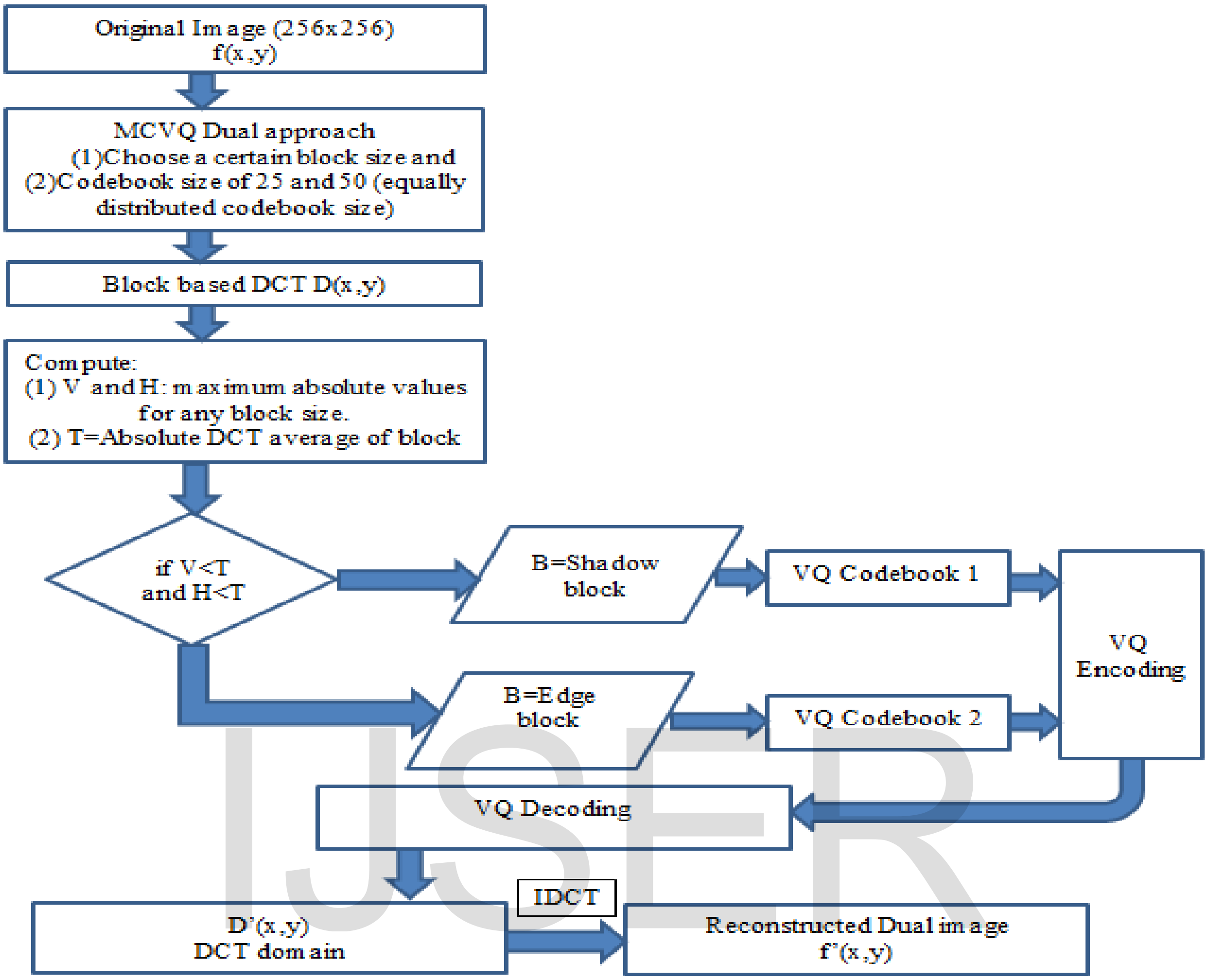

Figure 17. MCVQ Dual approach in DCT domain

7 EXPERIMENTAL RESULTS

The following are the experimental data of the techniques used in this research paper.

Let us look at the coefficients from our data in an example. Bridge image for 4x4 in spatial domain is:

151	161	161	164
151	154	159	159
153	151	158	159
153	153	154	158

Bridge image for 4x4 in DCT domain is:

112.7500	-12.2114	-0.7500	-0.0832
6.4768	-3.4445	-4.4048	-2.3410
2.7500	0.1817	-0.7500	-4.1342
1.9174	-0.8410	0.4716	0.4445

There is one DC coefficient which is at upper left corner with higher energy of 112.75, and there are fifteen AC coefficients. The negative AC coefficient has lower energy, while the positive AC coefficient has higher energy. A ranking of DCT coefficients is listed in Figure 18.

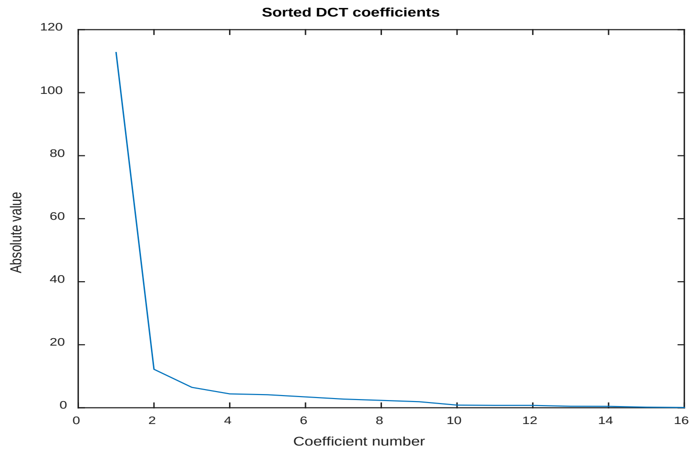

Figure 18. Ranked DCT coefficients with absolute values of Bridge image for 4x4 block

7.1 Results of standard VQ in DCT domain. Table 12 shows the Bridge images of K-Means using VQ in DCT domain. The detailed information in the Table 12 represents different block size used as 4x4 (16) and 8x8 (64) with codebook size of 25 and 50. By using these block and codebook sizes, different performance metrics have been calculated to check the quality of the Bridge image.

Table 1.
Bridge image of standard VQ using K-Means in DCT domain

Image 256x256	Bit Rate	Compression Ratio (CR)	SNR	PSNR	MSE	Execution Time
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

The block size of 16 and 64 and codebook size of 25 and 50 have been performed for MCVQ Dual approach in terms of PSNR, execution time, etc. In all scenarios of K-Means using VQ (standard VQ for single codebook) approach in DCT domain, block size of 16 and codebook size of 50 shows one of the best Bridge image quality results with higher PSNR and less execution time. The block size of 16 and codebook size of 50 for Bridge image with higher PSNR of 30.85 (dB) and execution time of 30.66 seconds as comparing to the other scenarios. It is evident that higher execution time due to a larger number of blocks (4096 blocks) used as compared to other parameters.

The following are the experimental data of the techniques employed in this chapter 4.

7.2 Results of MCVQ Quadruple approach in DCT domain. MCVQ Quadruple Codebook size of 25 is divided into four classifiers such as 6 of each. Codebook size 1 of 6 is set for Shadow, Codebook size 2 of 6 is set for Diagonal edge, Codebook size 3 of 6 is set for Horizontal edge, and Codebook size 4 of 6 is set for Vertical edge.

MCVQ Quadruple Codebook size of 50 is divided into four classifiers such as 12 of each. Codebook size 1 of 12 is set for Shadow, Codebook size 2 of 12 is set for Diagonal edge, Codebook size 3 of 12 is set for Horizontal edge, and Codebook size 4 of 12 is set for Vertical edge.

The block size of 16 and 64 and codebook size of 25 and 50 have been performed for MCVQ Quadruple approach in terms of PSNR, execution time, etc. In all scenarios of MCVQ Quadruple approach, block size of 16 and codebook size of 50 shows one of the best Bridge image quality results with higher PSNR and less execution time. The block size of 16 and codebook size of 50 for Bridge image with higher PSNR of 32.95 (dB) and execution time of 25.71 seconds as comparing to the other scenarios.

Table 2.
Bridge image of MCVQ Quadruple using K-Means in DCT domain

Image 256x256	Bit Rate	Compression Ratio (CR)	SNR	PSNR	MSE	Execution Time
64x25	0.0726	110.2532	25.991	28.456	1780.34	5.524
64x50	0.0882	90.7181	26.795	29.788	1719.55	5.668
16x25	0.2902	27.5633	26.813	31.045	952.86	25.605
16x50	0.3527	22.6795	27.985	32.950	940.27	25.711

7.3 Results of MCVQ Dual approach in DCT domain. MCVQ Dual Codebook size of 25 is divided into two classifiers such as 12 of each. Codebook size 1 of 12 is set for Shadow and Codebook size 2 of 12 is set for Edge.

MCVQ Dual Codebook size of 50 is divided into two classifiers such as 25 of each. Codebook size 1 of 25 is set for Shadow and Codebook size 2 of 25 is set for Edge,

The block size of 16 and 64 and codebook size of 25 and 50 have been performed for MCVQ Dual approach in terms of PSNR, execution time, etc. In all scenarios of MCVQ Dual approach, block size of 16 and codebook size of 50 shows one of the best Bridge image quality results with higher PSNR and less execution time. The block size of 16 and codebook size of 50 for Bridge image with higher PSNR of 34.291 (dB) and execution time of 24.75 seconds as comparing to the other scenarios.

Table 3.
Bridge image of MCVQ Dual using K-Means in DCT domain

Image 256x256	Bit Rate	Compression Ratio (CR)	SNR	PSNR	MSE	Execution Time
64x25	0.0726	110.2532	27.787	30.324	1446.28	4.641
64x50	0.0882	90.7181	28.458	31.669	1298.46	5.264
16x25	0.2902	27.5633	28.758	33.492	918.46	23.917
16x50	0.3527	22.6795	29.663	34.291	909.33	24.748

7.4 PSNR and Execution time comparison between standard VQ and MCVQ approaches in DCT domain. PSNR comparison between the standard VQ, and MCVQ Quadruple/Dual approaches has been performed for the Bridge image of block size of 16 and 64 with codebook size of 25 and 50 in Table 4.

Table 4.
PSNR comparison results between standard VQ and MCVQ approaches in DCT domain

Image 256x256	16x25	16x50	64x25	64x50
Standard VQ	28.312	30.853	26.658	27.644
MCVQ Quad	31.045	32.95	28.456	29.788
MCVQ Dual	33.492	34.291	30.324	31.669

Figure 19 shows the Table 4 results for Bridge image as:

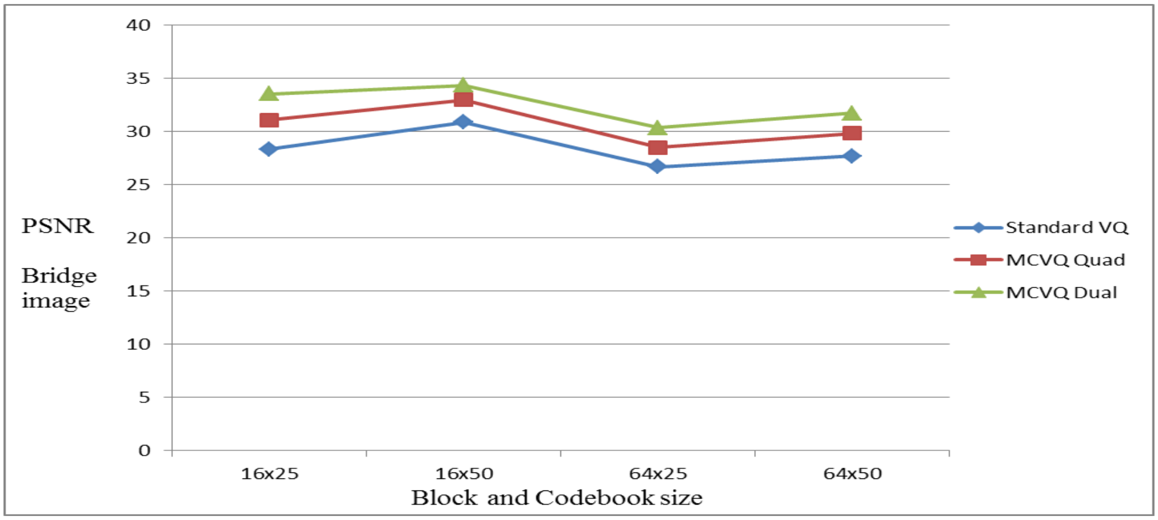

Figure 19. PSNR comparison results between standard VQ and MCVQ approaches in DCT domain

Execution time comparison between the standard VQ, and MCVQ Quadruple/Dual approaches has been performed for the Bridge image of block size of 16 and 64 with codebook size of 25 and 50 in Table 5.

Table 5.

Execution time comparison results between standard VQ and MCVQ approaches in DCT domain

Image 256x256	**16x25**	**16x50**	**64x25**	**64x50**
Standard VQ	29.319	30.6587	10.4408	10.527
MCVQ Quad	25.605	25.711	5.524	5.668
MCVQ Dual	23.917	24.748	4.641	5.264

Figure 20 shows the Table 5 results for Bridge image as:

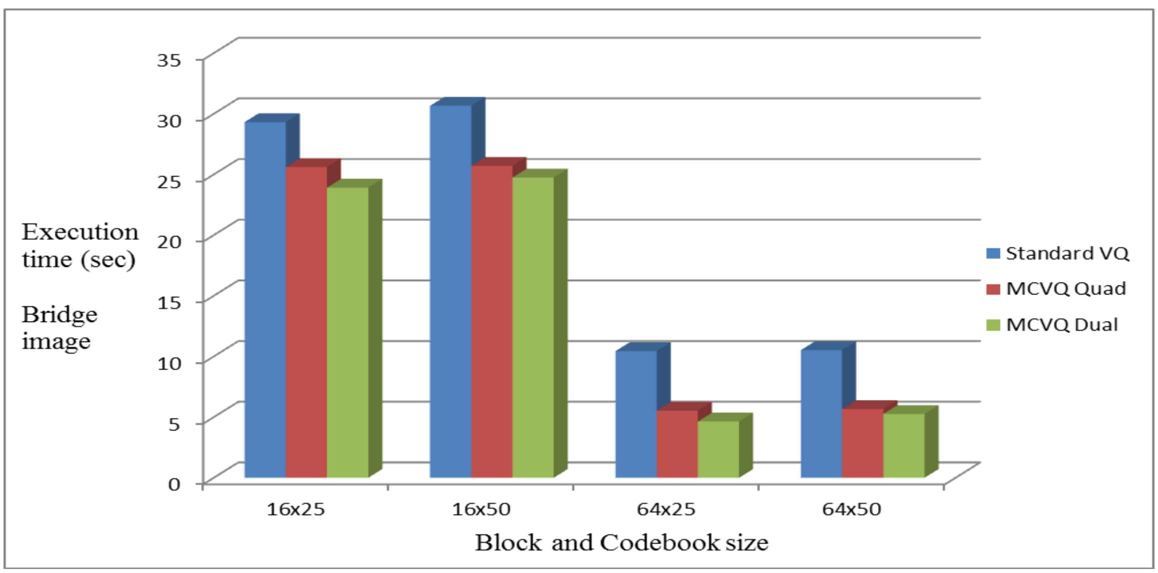

Figure 20. Execution time comparison results between standard VQ and MCVQ approaches in DCT domain

We can say from Table 4-5 and Figures 19-20 that MCVQ Dual approach takes less execution time for higher PSNR value than other approaches

Figures 21 (a)-(f) represents the reconstructed Bridge image of block sizes of 16 and 64 and codebook size of 50 for standard VQ method and MCVQ approach in DCT domain. Figure 21 (a) is the reconstructed standard VQ method for block size of 16 and codebook size of 50 in DCT domain. Figure 21 (b) is the reconstructed MCVQ Quadruple Bridge image for block size of 16 and codebook size of 50 in DCT domain. Figure 21 (c) is the reconstructed MCVQ Dual Bridge image for block size of 16 and codebook size of 50 in DCT domain. Figure 21 (d) is the reconstructed standard VQ method for block size of 64 and codebook size of 50 in DCT domain. Figure 21 (e) is the reconstructed MCVQ Quadruple Bridge image for block size of 64 and codebook size of 50 in DCT domain. Figure 21 (f) is the reconstructed MCVQ Dual Bridge image for block size of 64 and codebook size of 50 in DCT domain.

Figure 21. Bridge image of various block size with codebook size of 50 in DCT domain

By analyzing the results of standard VQ and MCVQ Quadruple/Dual approach, block size of 16 and codebook size of 50 is one of the best scenario to show better results in Figure 21(c) with higher PSNR, higher SNR and with less execution time. If we look at the Table results (Table 1, Table 2, Table 3), standard VQ Bridge image PSNR is 30.85 (dB), MCVQ Quadruple Bridge image PSNR is 32.95 (dB) and MCVQ Dual Bridge image PSNR is 34.29 (dB) for block size of 16 and codebook size of 50. MCVQ Dual approach has more than 2 (dB) differences between the standard VQ and MCVQ Quadruple approach in terms of PSNR. Again, if we look at the Table results, (Table 1, Table 2, Table 3), standard VQ Bridge image execution time is 30.66 seconds, MCVQ Quadruple Bridge image execution time is 25.71 seconds, and MCVQ Dual Bridge image execution time is 24.75 seconds.

Finally, Adaptive MCVQ Dual approach is the best approach for Bridge image results. Overall, MCVQ Dual approach outperforms one of the best approach with higher PSNR of 34.29 (dB) and less execution time of 24.75 seconds for Bridge image (block size of 16 and codebook size of 50) than standard VQ and MCVQ Quadruple approach in DCT domain.

8 PERFORMANCE METRICS

There are following performance metrics used for these standard VQ and MCVQ approaches.

8.1 Lossy compression.

(a) Bit rate is defined as:

$$\text{Bit Rate} = \frac{\log_2 N}{M} \quad \textbf{(6)}$$

'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

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

(c) MSE. MSE (Mean Square Error)

($Y_{i,j}$) is the reconstructed image and the ($X_{i,j}$) is the original image. **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. It is given by

$$\text{MSE} = \frac{\sum_{i=0}^{m-1}\sum_{j=0}^{n-1}[|\,Y_{(i,j)} - X_{(i,j)}|^2]}{(mxn)} \quad \textbf{(8)}$$

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

$$\text{SNR} = \frac{10*\log_{10}*(\sum_{i=0}^{m-1}\sum_{j=0}^{n-1}X_{(i,j)}^2)}{\sum_{i=0}^{m-1}\sum_{j=0}^{n-1}[Y_{(i,j)} - X_{(i,j)}]^2} \quad \textbf{(9)}$$

(e) PSNR.

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

1. Conclusion and Future work:

The absolute volume of data and its growth in the future are considered as the most important and key challenges in Big Data Technology. High dimensional nature of data is experienced in most of these applications.

We proposed to implement VQ techniques on the frequency domain, specifically DCT. Because of its energy packing property, DCT gives better image quality and less execution time.

The research also proposed to investigate the further advantages of working in the DCT domain. The MCVQ technique then was adapted. It takes an advantage of the property of good energy-compaction to simplify the block classification problem. MCVQ Quadruple technique was implemented first where we classify each block with name listed as Shadow, Horizontal edge, Vertical edge, and Diagonal edge. So, this technique partitions AC non-zero blocks into four equal-sized codebooks. This technique also performs better than the standard VQ method used in DCT domain. Also MCVQ Dual techniques were then implemented where we classify each block with name listed as Shadow and Edge. MCVQ Dual technique partitions AC non-zero blocks into two equal-sized codebooks. MCVQ Dual technique results in one of the best quality images and less execution time for block size of 16 and codebook size of 50 than all MCVQ Quadruple approach and the standard VQ for single codebook in DCT domain.

It meets the conditions of good-quality reconstructed image with higher PSNR and higher SNR for a given acceptable CR, and less MSE and the less execution time.

Future research is to analyze and to implement image compression VQ techniques in other clustering algorithms and other frequency domains.

Acknowledgement:

I want to thanks Dr. Jung H. Kim and Dr. Robert Li for his support and contribution along with his ideas in this research.

References

[1] Michael K, Miller KW. Big data: new opportunities and new challenges. *Editorial: IEEE Computer*. 2013; 46(6):22–24.

[2] Begoli E, Horey J. Design principles for effective knowledge discovery from big data. Proceedings of the 10th Working IEEE/IFIP Conference on Software Architecture (ECSA '12); August 2012; pp. 215–218.

[3] Talia D. Clouds for scalable big data analytics. *Computer*. 2013; 46(5):98–101.6515548

[4] Shukla R., Gupta N. Kumar: Image Compression through DCT and Huffman Coding Technique. International Journal of Current Engineering and Technology Vol.5, No.3, June 2015, pp. 1942-1946.

[5] A. Gersho and R. Gray, *Vector Quantization and Signal Compression*, Kluwer Academic Publisher, 1992.

[6] A. K. Jain, image data compression: a review," *Proc. IEEE*, vol. 69, pp. 349-389.

[7] 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.

[8] T. Chiueh, T. Tang, and L. Chen (1994) Vector quantization using tree-structured self-organizing feature maps, *IEEE Journal on Selected Areas in Communications*, Vol. 12, No. 9, pp.1594-1599.

[9] Tseng, H-W., and C-C. Chang. "A very low bit rate image compressor using transformed classified vector quantization." *Informatica* 29.3 (2005).

[10] G. K. Wallace, “The JPEG still picture compression standard,” Communications of ACM –Digital Multimedia Systems, vol. 34, no. 4, pp. 30-44, Apr 1991.
[11] Robert Li, Earnest Sherrod, Jung H. Kim and Gao Pan, “Fast Image Vector Quantization using a Modified Competitive Learning Neural Network Approach", The International Journal of Imaging System and Technology, vol.8, pp413-418, 1997.
[12] Robert Li, Jung Kim and N. Al-Shamakhi, “*Image Compression Using Transformed Vector Quantization*”, Image Vision Computing, vol. 20, no. 1, pp37-45, January 2002.
[13] O. Rioul and M. Vetterli, “Wavelets and Signal Processing”, IEEE Signal Process. Mag. Vol 8, pp. 14-38, Oct.1991

Biography

Dr. Ali Tariq Bhatti had his MSc. in Electrical Engineering from North Carolina A&T State University. He had his PhD in Electrical Engineering from North Carolina A&T State University. His research interest includes Power & System Control, Signal, Image and Video processing, Wireless & Telecommunication, Biosensors, and Software Development.