Multiwavelet Reconstruct Image Enhancement using 1 and 2 Order Approximations Based on Multi-Stage Vector Quantization

Adil Abdulwahhab Ghidan Al-Azzawi and Muneera Abed Hmdi Al-Saedi

Diyala University/College of Science/Computer Science Department, Iraqi Commission for Computers & Informatics / Institute for Postgraduate Studies

Receiving Date: 15-03-2011 **- Accept Date:** 14-06-2011

Abstract

One from the most problems of multiwavelet transform is listing by the existing image coding standards generally degrades at low bit-rates and lose some of data after applying image reconstruction because of the underlying block based. Due to implementation constraints multiwavelets do not posses all the properties such as orthogonally, short support, linear phase symmetry, and a high order of approximation through vanishing moments simultaneously, which are very much essential for signal processing. New class of wavelets called 'Multiwavelets' which possess more than one scaling function overcomes this problem. This paper presents a new image technique scheme based on reconstruct approximation of Multiwavelets coefficients along with multistage vector quantization named as Multi Stage Vector Quantization using the inverse computation of multiwavelet transform, after apply the multi-stage vector quantization named as (MSVQ-IDMWT). The performance of the proposed scheme is compared with the results obtained from regular technique of reconstructing (IDMWT) from the SNR and PSNR values.

Introduction

Digital representation of image has created the need to transform the image from one domain to other that transformed used in digital image processing applications, observed image is modeled to be corrupted by different types of noise that results in a noisy version. Hence, image denoising is an important problem that aims to find an estimate version from the noisy image that is as close to the original image as possible, or to efficient compression algorithms that will reduce the storage space and the associated channel bandwidth for the transmission of images. Transform aims at changing the representation of a signal or a function by using of a mathematical operation. It is possible also to decompose a complex problem into simpler ones for obtaining simpler solution. Transforms play an important role in different signal processing applications like filtering, pattern recognition, restoration, spectrum estimation, signal enhancement, localization and compression [1]. The performance of each application depends on several factors, and hence, each application may need a different transform technique for a better solution [2].Multiwavelet is one type of the image transform that depended on the filter banks require a vector-value input signal. There are a number of ways to produce such a signal from 2-D signal image data. Perhaps the most obvious method is to use adjacent row and columns of the image data [3]. However, these approach doses not work well for general multiwavelets and leads to reconstruction artifacts in the lowpass data after coefficient quantization [4]. This problem can be avoided by constructing "constrained" multiwavelets, which possess certain key properties. Unfortunately, the extra constraints are somewhat restrictive; image compression test show that constrained multiwavelets do not perform as well as some other multifilters [5]. Another approach is to first split each row or column into two half-length signal, and then use these two half signal as the channel inputs into the multifilter. A naïve approach, as Strela points out [6], is to simply take the odd samples for one signal and the even samples for the second signal.

Multiwavelet Transform

In multiwavelet transform, we use multiwavelet as transform basis. Multiwavelet functions are functions generated from one single function ψ by scaling and translation:

$$\psi_{a,b}(t)= \frac{1}{\sqrt{a}}\,\psi_a\left(\frac{t-b}{a}\right) \qquad \dots (1)$$

The mother wavelet $\psi\,(t)$ has to be zero integral, $\int \psi_{a,b}(t)dt = 0$. From (1) we see that high frequency multiwavelet correspond to $a > 1$ or narrow width, while low frequency multiwavelet corresponds to $a < 1$ or wider width. The basic idea of wavelet transform is to represent any function f as a linear superposition of wavelets. Any such superposition decomposes f to different scale levels, where each level can be then further decomposed with a resolution adapted to that level. One general way to do this is writing f as the sum of wavelets $\psi_{m,n}(t)$over m and n . This leads to discrete wavelet transform [7]:

$$f(t) = \sum_{m,n} \psi_{m,n}(t) \qquad \dots (2)$$

By introducing the multi-resolution analysis (MRA) idea by Mallat [3], in discrete wavelet transform we really use two functions: wavelet function $\psi(t)$ and scaling function $\varphi(t)$. If we have a scaling function $\varphi(t)\in L^2(R)$, then the sequence of subspaces spanned by its scaling and translations $\psi_{j,k}(t)=2^{j/2}\,\varphi(2^j\,t - k)$, i.e

$$Vj = span\{\varphi j,k\,(t), j,k \in Z\} \qquad \dots (3)$$

Constitute a MRA for $L^2(R)$.

$\varphi(t)$ must satisfy the MRA condition [7]:

$$\varphi(t)=\sqrt{2}\sum h(n)\,\varphi(2t-2) \qquad \dots (4)$$

For $n\in Z$. In this manner, we can span the difference between spaces V_j by wavelet functions produced from mother wavelet: $\psi_{j,k}(t)=2^{j/2}\,\varphi(2^j\,t - k)$ Then we have [8]:

$$\psi_{j,k}(t) = \sqrt{2}\sum g(n)\,\varphi(2t-2) \qquad \dots (5)$$

For orthogonal basis we have [8]:

$$g(n) = (-1)^n\,h(-n+1) \qquad \dots (6)$$

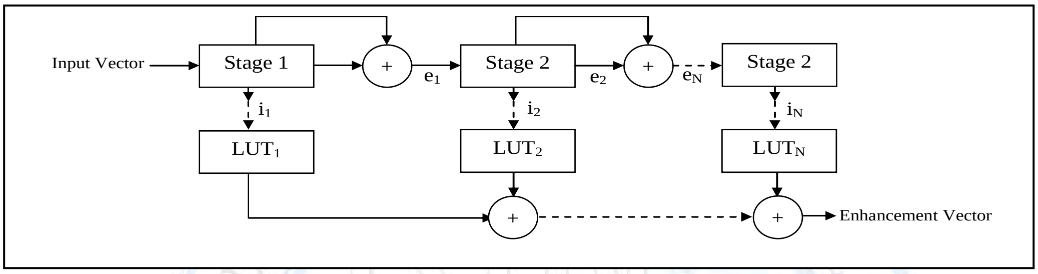
If we want to find the projection of a function $f(t) \in L^2(R)$ on this set of subspaces, we must express it in e as a linear combination of expansion functions of that subspace [4]:

$$f(t) = \sum_{n=-\infty}^{\infty} c(t)\ \varphi(t) + \sum_{n=-\infty}^{\infty} \sum_{n=-\infty}^{\infty} d(t)\ \psi_{j,k} \qquad \dots (7)$$

Where $\varphi_k(t)$ corresponds to the space V_0 and $\psi_{j,k}(t)$ corresponds to wavelet spaces.By using the idea of MRA implementation of wavelet decomposition can be performed using filter bank constructed by a pyramidal structure of lowpass filters $h(n)$ and highpass filters $g(n)$ [9, 10].

Multi-Stage vector Quantization

Vector quantization is a powerful tool for data reduction. Vector quantization extends scalar quantization to higher dimensional space. By grouping input samples into vectors and using a vector quantizer, a lower bit rate and higher performance can be achieved. However, the codebook size and the computational complexity increase exponentially as the rate increases for a given vector size. Full-search VQ such as entropy-constrained VQ (ECVQ) enjoys small quantization distortion. However, it has long compression time, and may not be well suited for real time signal compression systems. Tree-structured VQ (TSVQ) although can significantly reduce the compression time, has the disadvantage that the storage size required for the VQ is usually very large and cannot be controlled during the design process. Therefore, it may not be convenient to use TSVQ for the applications where the storage size is a major concern. A structured VQ scheme which can achieve very low encoding and storage complexity is MSVQ [12]. This appealing property of MSVQ motivated us to use MSVQ in the quantization stage. The basic idea of multistage vector quantization is to divide the encoding task into successive stages, where the first stage performs a relatively crude quantization of the input vector. Then a second-stage quantizer operates on the error vector between the original and the quantized first-stage output. The quantized error vector then provides a second approximation to the original input vector thereby leading to a refined or more accurate representation of the input. A third stage quantizer may then be used to quantize the second-stage error to provide a further refinement and so on. In this paper, we have implemented two-stage vector quantizer. The input vector is quantized by the initial or first stage vector quantizer denoted by VQ1 whose code book is $C_1 = C_{10}, C_{11} ...C_1 (N_{1-1})$ with size N_1. The quantized approximation ˆx1 is then subtracted from x producing the error vector. This error vector is then applied to a second vector quantizer VQ2 whose code book is $C_2 = C_{20}, C_{21}... C_2 (N_{2-1})$ with size N2 yielding the quantized output [10].



Figure(1): Multistage Vector Quantization System

The encoder transmits a pair of indices specifying the selected codeword for each stage and the task of the decoder is to perform two table lookups to generate and then sum the two code words. In fact, the overall codeword or index is the concatenation of code words or indices chosen from each of two codebooks. Thus, the equivalent product codebook can be generated from the Cartesian product $C_1 \times C_2$. Compared to the full-search VQ with the product codebook C, the two stage VQ can reduce the complexity from $N = N_1 \times N_2$ to $N_1 + N_2$. The multistage vector quantization system for 'N' stages is shown in Fig. 2. In the figure, 'X' represents the input vector, LUT stands for lookup table and i_1 , i_2 , etc represent indices from different stages. The overall index is the concatenation of indices chosen from each of the two codebooks. From the Fig.2, it is evident that the input vector is given only to the first stage, whereas the input to the successive stages is the error vectors from the previous stage which are denoted by N ....e, e 2 1. Xˆ is the reconstructed signal at the decoder end [11].

Computing Inverse DMWT Using Critically- Sampled Scheme of Preprocessing using Multistage Vector Scalar (MSV-QIDMWT)

A general procedure can be followed for computing a single-level 2-D discrete multiwavelets inverse transform by using GHM four multifilters and using a critically-sampled scheme of Postprocessing (approximation-based scheme of Postprocessing). By using a critically-sampled scheme of Postprocessing (approximation-based scheme of Postprocessing); the DMWT matrix has the same dimensions as compared with that of the original matrix which should be a square N×N matrix where N should be power of 2. So, to reconstruct the original N×N matrix, a reconstruction matrix, which is the inverse (or transpose) of transformation matrix given (2.2), dimensions should be equal to critical-sampled preprocessing DMWT N×N matrix dimensions. As there are two order or approximation types of critically-sampled preprocessing, 1st order and 2nd order approximations, there are correspondingly two types of critically-sampled Postprocessing methods that should be followed; one for each order of approximations.To compute a single-level 2-D Inverse Discrete Multiwavelets Transform using critically-sampled scheme of Postprocessing, the next steps should be followed:

1. ***Coefficient Shuffling***, which is applied to the DMWT N×N matrix four basic subbands individually. For each subbands, coefficient shuffling shuffles the columns first then shuffles the rows.
2. ***Column Reconstruction***,
   - **I.** Transpose the coefficient shuffled N×N matrix.
   - **II.** Apply shuffling by arranging the row pairs 1, 2 and 3, 4… (N/2)-1, (N/2) of the coefficient shuffled N×N matrix transpose to be the row pairs 1, 2 and 5, 6… N-3, N-2 of the resulting matrix and arranging the row pairs (N/2)+1, (N/2)+2, and (N/2)+3, (N/2)+4… N-1, N of the coefficient shuffled N×N matrix transpose to be the row pairs 3, 4 and 7, 8… N-1, N of the resulting matrix.
   - **III.** Multiply an N×N reconstruction matrix (N×N transformation matrix transpose) with the resulting N×N shuffled matrix from II.
3. ***Postprocessing***, a critical-sampled scheme of Postprocessing can be computed as follows:

I.1st order approximation Postprocessing: can be computed by applying these equations:

Odd-row= [[same odd-row] - (0.11086198) [next even-row]

- (0.11086198) [previous even-row]] / (0.373615) … *(8)*

Even-row= [same even-row] / (√2 -1) … *(9)*

to the odd- and even-rows of the column reconstructed N×N matrix respectively.

II.2nd order approximation Postprocessing: can be computed by applying these equations:

Odd-row= [[same odd-row] - (3/8√8) [next even-row]

- (3/8√2) [previous even-row]]/ (10/8√2) … *(10)*

Even-row= [same even-row] … *(11)*

to the odd- and even-rows of the column reconstructed N×N matrix respectively.

4. ***Row*** **and** ***Column reconstruction***

I.Transpose the Postprocessed N×N resultant matrix.

II. Apply shuffling by arranging the row pairs 1, 2 and 3, 4… (N/2)-1, N/2 of the N×N matrix Postprocessed resultant matrix transpose to be the row pairs 1, 2 and 5, 6… N-3, N-2 of the resulting matrix and arranging the row pairs (N/2) +1, (N/2) +2, and (N/2) +3, (N/2)+4… N-1, N of the N×N matrix Postprocessed resultant matrix transpose to be the row pairs 3, 4 and 7, 8… N-1, N of the resulting matrix.

III. Apply the ***Multistage Vector Quantization*** for row preprocessing to the input 2-D matrix, *X*, using repeated row preprocessing and c= c10, c11 ...c1 (N1−1)

$$X = \begin{bmatrix} x_{0,0} & x_{0,1} & x_{0,2} & x_{0,3} \\ x_{1,0} & x_{1,1} & x_{1,2} & x_{1,3} \\ x_{2,0} & x_{2,1} & x_{2,2} & x_{2,3} \\ x_{3,0} & x_{3,1} & x_{3,2} & x_{3,3} \end{bmatrix} \xrightarrow[\text{rows}]{\text{preprocess}} x = \begin{bmatrix} x_{0,0} & x_{0,1} & x_{0,2} & x_{0,3} \\ cx_{0,0} & cx_{0,1} & cx_{0,2} & cx_{0,3} \\ x_{1,0} & x_{1,1} & x_{1,2} & x_{1,3} \\ cx_{1,0} & cx_{1,1} & cx_{1,2} & cx_{1,3} \\ x_{2,0} & x_{2,1} & x_{2,2} & x_{2,3} \\ cx_{2,0} & cx_{2,1} & cx_{2,2} & cx_{2,3} \\ x_{3,0} & x_{3,1} & x_{3,2} & x_{3,3} \\ cx_{3,0} & cx_{3,1} & cx_{3,2} & cx_{3,3} \end{bmatrix} \quad \ldots (12)$$

1. Apply row transformation

I. Let $[z]= [W]\times [x]$

II. Permute $[z]$,

$$z = \begin{bmatrix} z_{0,0} & z_{0,1} & z_{0,2} & z_{0,3} \\ z_{1,0} & z_{1,1} & z_{1,2} & z_{1,3} \\ z_{2,0} & z_{2,1} & z_{2,2} & z_{2,3} \\ z_{3,0} & z_{3,1} & z_{3,2} & z_{3,3} \\ z_{4,0} & z_{4,1} & z_{4,2} & z_{4,3} \\ z_{5,0} & z_{5,1} & z_{5,2} & z_{5,3} \\ z_{6,0} & z_{6,1} & z_{6,2} & z_{6,3} \\ z_{7,0} & z_{7,1} & z_{7,2} & z_{7,3} \end{bmatrix} \xrightarrow{\text{permute}} p = \begin{bmatrix} p_{0,0} & p_{0,1} & p_{0,2} & p_{0,3} \\ p_{1,0} & p_{1,1} & p_{1,2} & p_{1,3} \\ p_{4,0} & p_{4,1} & p_{4,2} & p_{4,3} \\ p_{5,0} & p_{5,1} & p_{5,2} & p_{5,3} \\ p_{2,0} & p_{2,1} & p_{2,2} & p_{2,3} \\ p_{3,0} & p_{3,1} & p_{3,2} & p_{3,3} \\ p_{6,0} & p_{6,1} & p_{6,2} & p_{6,3} \\ p_{7,0} & p_{7,1} & p_{7,2} & p_{7,3} \end{bmatrix}, \quad \ldots (13)$$

2. Apply column transformation

I. Transpose p matrix.

$$p^t = \begin{bmatrix} p_{0,0} & p_{1,0} & p_{4,0} & p_{5,0} & p_{2,0} & p_{3,0} & p_{6,0} & p_{7,0} \\ p_{0,1} & p_{1,1} & p_{4,1} & p_{5,1} & p_{2,1} & p_{3,1} & p_{6,1} & p_{7,1} \\ p_{0,2} & p_{1,2} & p_{4,2} & p_{5,2} & p_{2,2} & p_{3,2} & p_{6,2} & p_{7,2} \\ p_{0,3} & p_{1,3} & p_{4,3} & p_{5,3} & p_{2,3} & p_{3,3} & p_{6,3} & p_{7,3} \end{bmatrix} \quad \ldots (14)$$

II. preprocess $[p]^t$ to get $[P]$ matrix

$$P = \begin{bmatrix} P_{0,0} & P_{1,0} & P_{4,0} & P_{5,0} & P_{2,0} & P_{3,0} & P_{6,0} & P_{7,0} \\ \alpha P_{0,0} & \alpha P_{1,0} & \alpha P_{4,0} & \alpha P_{5,0} & \alpha P_{2,0} & \alpha P_{3,0} & \alpha P_{6,0} & \alpha P_{7,0} \\ P_{0,1} & P_{1,1} & P_{4,1} & P_{5,1} & P_{2,1} & P_{3,1} & P_{6,1} & P_{7,1} \\ \alpha P_{0,1} & \alpha P_{1,1} & \alpha P_{4,1} & \alpha P_{5,1} & \alpha P_{2,1} & \alpha P_{3,1} & \alpha P_{6,1} & \alpha P_{7,1} \\ P_{0,2} & P_{1,2} & P_{4,2} & P_{5,2} & P_{2,2} & P_{3,2} & P_{6,2} & P_{7,2} \\ \alpha P_{0,2} & \alpha P_{1,2} & \alpha P_{4,2} & \alpha P_{5,2} & \alpha P_{2,2} & \alpha P_{3,2} & \alpha P_{6,2} & \alpha P_{7,2} \\ P_{0,3} & P_{1,3} & P_{4,3} & P_{5,3} & P_{2,3} & P_{3,3} & P_{6,3} & P_{7,3} \\ \alpha P_{0,3} & \alpha P_{1,3} & \alpha P_{4,3} & \alpha P_{5,3} & \alpha P_{2,3} & \alpha P_{3,3} & \alpha P_{6,3} & \alpha P_{7,3} \end{bmatrix} \quad \ldots (15)$$

III. let $[b]= [W] \times [P]$ … (16)

IV. Permute $[b]$ to get $[B]$matrix which is 8×8 matrix.

V.

$$b = \begin{bmatrix} b_{0,0} & b_{0,1} & b_{0,2} & b_{0,3} & b_{0,4} & b_{0,5} & b_{0,6} & b_{0,7} \\ b_{1,0} & b_{1,1} & b_{1,2} & b_{1,3} & b_{1,4} & b_{1,5} & b_{1,6} & b_{1,7} \\ b_{2,0} & b_{2,1} & b_{2,2} & b_{2,3} & b_{2,4} & b_{2,5} & b_{2,6} & b_{2,7} \\ b_{3,0} & b_{3,1} & b_{3,2} & b_{3,3} & b_{3,4} & b_{3,5} & b_{3,6} & b_{3,7} \\ b_{4,0} & b_{4,1} & b_{4,2} & b_{4,3} & b_{4,4} & b_{4,5} & b_{4,6} & b_{4,7} \\ b_{5,0} & b_{5,1} & b_{5,2} & b_{5,3} & b_{5,4} & b_{5,5} & b_{5,6} & b_{5,7} \\ b_{6,0} & b_{6,1} & b_{6,2} & b_{6,3} & b_{6,4} & b_{6,5} & b_{6,6} & b_{6,7} \\ b_{7,0} & b_{7,1} & b_{7,2} & b_{7,3} & b_{7,4} & b_{7,5} & b_{7,6} & b_{7,7} \end{bmatrix}$$

Permute ↓

$$B = \begin{bmatrix} B_{0,0} & B_{0,1} & B_{0,2} & B_{0,3} & B_{0,4} & B_{0,5} & B_{0,6} & B_{0,7} \\ B_{1,0} & B_{1,1} & B_{1,2} & B_{1,3} & B_{1,4} & B_{1,5} & B_{1,6} & B_{1,7} \\ B_{4,0} & B_{4,1} & B_{4,2} & B_{4,3} & B_{4,4} & B_{4,5} & B_{4,6} & B_{4,7} \\ B_{5,0} & B_{5,1} & B_{5,2} & B_{5,3} & B_{5,4} & B_{5,5} & B_{5,6} & B_{5,7} \\ B_{2,0} & B_{2,1} & B_{2,2} & B_{2,3} & B_{2,4} & B_{2,5} & B_{2,6} & B_{2,7} \\ B_{3,0} & B_{3,1} & B_{3,2} & B_{3,3} & B_{3,4} & B_{3,3} & B_{3,6} & B_{3,7} \\ B_{6,0} & B_{6,1} & B_{6,2} & B_{6,3} & B_{6,4} & B_{6,5} & B_{6,6} & B_{6,7} \\ B_{7,0} & B_{7,1} & B_{7,2} & B_{7,3} & B_{7,4} & B_{7,5} & B_{7,6} & B_{7,7} \end{bmatrix} \quad \ldots (17)$$

3. The final **DMWT** matrix [Y] results from the following steps,

I. Transpose [B] matrix to get [y] matrix.

II. Apply coefficients permutation to each of the four basic sub bands of matrix [y].

5. ***Post processing,*** a critical-sampled scheme of Post processing can be done by the same process of step 3 above which results in the N×N original reconstructed 2-D signal matrix.

Proposed Algorithm

The proposed image coder scheme is explained below.

1. The correlation present in the input image is removed by taking multiwavelet transform of the input image.

2. The non-linear approximation of the multiwavelet coefficients is performed.

3. The transform coefficients obtained in step 2 are vector quantized in a multistage manner where the residual error coefficients due to quantization are iteratively feedback and vector quantized. If the number of stages in MSVQ is more, the refinement of the quantized vectors will be better. But it suffers from the need for a high bit- rate for each additional stage is added. Hence we have restricted our attention to two stages in Multistage Vector Quantization.

4. The outputs from step 3 are lossless coded using static Huffman code. This completes the encoder stage of the proposed algorithm which is illustrated in Fig. 2. In decoding, the decoder basically performs the reverse process of the above steps.

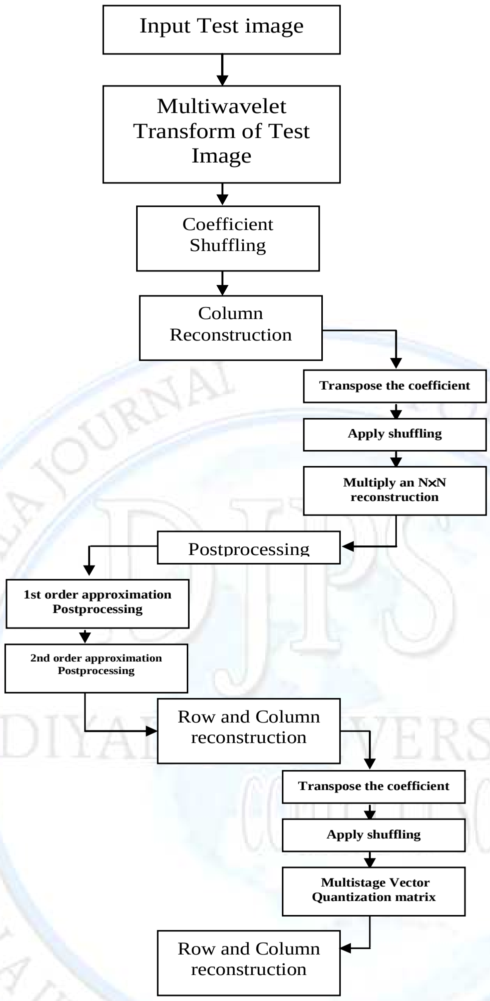


Figure (2): Block Diagram of proposed algorithm

Result and Discussion

We present the DMWT results for 256 X 256, 'Tiger' is class of natural image that do not contain large amounts of high-frequency or oscillating patterns. 'Tiger' image exhibits large amounts of high-frequency and oscillating patterns. 'Tiger' image contains significant amounts of both low and high-frequency region. The images are Reconstruct using multiwavelet transform. This reconstruct used in this experiment in figure 5 and figure 6. Shows the different method for computing MWDT using 1st and 2nd approximation then the reconstruction without using MVQ in IDMWT and with it. are GHM, filter. The results are compared with that of regular **IDMWT** and **MVQ-IDMWT** in the same way. Table 1, shows the comparative results of 'Tiger' images.

1. When the percentage of the significant coefficient retained is less, MSVQ-IMWT is giving better PSNR when compared to IDMWT.

2. As the bit rate increases, the PSNR value increases which is in accordance with Rate-Distortion theory [7]. Figure 4 shows the plot of PSNR against the percentage of coefficients retained for MSVQ-IMWT and IDMWT with rate as two, under second level of decomposition. When the SNR coefficients retained is less the gap between the performance of MSVQ-IMWT and scalar IDMWT is more and it merges with the increase in SNR coefficients. This is completely evident from the Fig. 4.

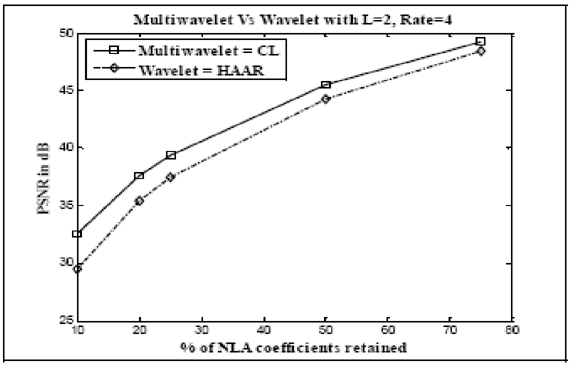


Figure (4): Comparison of PSNR values for IDMWT and MVSQ-IMWT for 'Tiger' image

Figure 5 shows the original and the reconstructed 'Tiger' image using MSVQ-IMWT and IDMWT transform with 10% of the coefficients retained with first level of decomposition and rate as four. From the Fig. 5b and 5c, it is obvious the visual quality of the reconstructed image using MSVQ-IMWT is better than that of IDMWT rather than table.1 describes the objective measurement of images with reconstruct using IDMWT and MVQ-IDMWT using first order approximated, table.2 shown the same measurement with second order approximation.

a. original Image   b. Reconstruct with IDMWT   b. Recon. With MSVQ-IDMWT

Figure (5): (a) Original image (b) Reconstructed image using IDMWT (c) Reconstructed image using MSVQ-IDMWT Depended on First Order Approximation

a. original Image   b. Reconstruct with IDMWT   b. Recon. With MSVQ-IDMWT

Figure (6): (a) Original image (b) Reconstructed image using IDMWT (c) Reconstructed image using MSVQ-IDMWT Depended on Second Order Approximation

Table (1): Objective measurement of images with reconstruct using IDMWT and MVQ-IDMWT using first order approximated

Original Image	Reconstruct IDMWT		Using MSVQ-IDMWT	
	PSNR	SNR	PSNR	SNR
1	20.6891	20.6532	21.2301	21.6621

Table (2): Objective measurement of images with reconstruct using IDMWT and MVQ-IDMWT using second order approximated

Original Image	Reconstruct IDMWT		Using MSVQ-IDMWT	
	PSNR	SNR	PSNR	SNR
1	20.787	20.321	22.0123	22.5021

Conclusion

In this work we have proposed a new image reconstruct algorithm based on multistage transform with vector quantization MSVQ. Our aim is to compare the performance of multi-stage vector quantization (MSVQ-IDMWT) agains t regular reconstruct multiwavelet using inverse discreet multiwavelet transform (IDMWT) on tiger images along with the application of multistage vector quantization on both the schemes. When the number of significant coefficients the performance of (MSVQ-IDMWT) this implies that, few significant multiwavelet coefficients are sufficient to reconstruct the image in a better manner than with the same significant regular multiwavelet reconstruction. If we allow more significant coefficients to the performance of regular dominates depending on using (MSVQ-IDMWT).

References

1. Michale B. Martin, *"Applications of multiwavelets to Image Compression"*, M.Sc. Thesis in Electrical Engineering, Virginia polytechnic Institute and State University(Virginia Tec),Blacksburg,june,2008.
2. Vasily Strela, Peter Niels Heter,Gilbert Strang, Pankaj Topiwala, and Christopher Heil,*" The Application of Multiwavelet Filterbank to Image Processing"*, IEE TRANSACTIONS ON IMAGE PROCESSING, VOL.8, NO.4,April 2009.
3. Michael B.Martin and Amy E.Bell, Member, IEE, "*New Image Compression Techniques Using Multiwavelet and Multiwavelet Packets*" IEE TRANSACTIONS ON IMAGE PROCESSING, VOL.10, NO.4 APRIL 2007.
4. Ifeachor, E. G., and Jervis, B. W., *“Digital signal processing”*. A practical approach, 5th printing, 2006.
5. Vasily Strela, Multiwavelets:, *"Theory and Applications"*, Ph.D. thesis, Massachusetts Institute of Technology,2006.
6. Tekapl, A.M., and Pavlovie, G., *“Digital image restoration”*, Springer-verlang, in A.K., Katsagglos ed., New York, U.S.A., 2008.
7. I. Daubechies, *"Tech Lectures on Wavelets"*, PP.(251),SLAM,2002.

8. Matheel, Emad, *"Color Image Denoising using Discrete Multiwavelet Transform"* Dep. Of Computer Science, University of Technology, 2004
9. J. Geronimo, D.Hardin, and P. R. Massopust, *"Fractal Function and Wavelet Expression on Several Function ",* J. Approx. Theory, vol.78,pp(373-401), 2004.
10. S. Santoso, E. J. Power, and W. M. Grady, “Power quality disturbance data compression usingwavelet transformmethods,” IEEE Trans. Power Delivery, vol. 12, pp. 1250–1256, July 2007.
11. J. Arrillaga,M. H. J. Bollen, and N. R.Watson, “Power quality following deregulation,” Proc. IEEE, vol. 88, pp. 246–261, Feb. 2000.