ISSN No. 0976-5697

Volume 2, No. 1, Jan-Feb 2011

International Journal of Advanced Research in Computer Science

RESEARCH PAPER

Available Online at www.ijarcs.info

K-Means Clustering and Wavelet Based Image Compression

E.Ponmalar
Department of Computer Science,
SFR College for women, Sivakasi, India
ponmalar2006@yahoo.co.in

T.Kathirvalavakumar*
Department of Computer Science
VHNSN College, Virudhunagar, India
kathirvalavakumar@yahoo.com

Abstract: This work proposes a novel method of performing image compression using a hybrid of K-means and wavelet techniques. K-means clustering technique is used as a vector quantizer to generate centroids when it is applied over the vectors representing an image. The proposed work compresses the centroids (code vectors) by applying the Discrete Wavelet Transform over the centroids. The approximation coefficients of the centroids along with the index values of the centroids corresponding to each input vector representing the image blocks form the compressed image. The proposed technique is implemented successfully and the experimental results show that the hybrid of K-means clustering and wavelet transform techniques produces better image compression with improved performance measures.

Keywords- Image Compression, K Means Clustering, Wavelet transform, DWT coefficients, decompression

I. INTRODUCTION

Clustering is a lossy data compression based on block coding. It can be directly used as a standalone technique for image compression. Clustering has its own advantages when compared with scalar quantization. Similarly, wavelets are one among the most popular and important techniques used in image compression.

The important idea of this proposed work is that the advantages of K-means clustering and the advantages of Discrete Wavelet Transform (DWT), a mathematical method are to be merged to form a hybrid technique for image compression. The proposed method achieves better image compression. The storage of the image is reduced and visual quality of the reconstructed image is maintained.

A. *K-Means Clustering Based Image Compression*

Vector quantization is a classical technique used for image compression. An N level vector quantization can be viewed as a mapping Q from a k dimensional Euclidean space R^k into a finite subset Y of R^k Q: $R^k \rightarrow$ Y where Y = $\{Y_1, Y2...Y_N\}$ is the set of reproduction vectors. In image vector quantization, image is partitioned into small equal blocks with k elements. For every k dimensional vector X, the codeword Y_i is computed such that Mean Square Error (MSE) between X_i and Y_i is minimum. Y_i is the centroid and Y the code book. Thus, quantization deals with creation of a code book consisting of code words. The goal of vector quantizer is to reduce the memory required to store the code words, as the code words and indexes form the compressed image. Clustering is such a quantization technique where mapping from larger subspace to smaller subspace is performed. The smaller subspace now requires less storage space. Many researches have been pursed in this area. Karayiannis[1] has formulated a generalized K-Means algorithm for image compression on vector quantization. Vereka and Buchanan [2] proposed a colour quantization using K-means algorithm where local K-means procedure is used to speed up the quantization. A new algorithm for vector quantization using modified K-means algorithm has been designed in the codebook updating step by Lee, Baek and Sung [4]. Implementation of K-means clustering algorithm over real data sets in color quantization, data compression and color segmentation has been done by Kanungo [5]. Fradkin, Muchnik, Streltsov[6] used divisive K-means clustering for image compression in military surveillance. Chou and Lai [7] proposed a K-means algorithm that is different from the conventional K-means in assigning more cluster centres to areas of low densities of data. Speeding up the K-means clustering algorithm for palette design is proposed by Hu and Su [8]. The performance of K-means as a quantizer for color images has been analysed by Celebi [9].

B. *Wavelet based Image Compression*

Use of wavelets for image compression is not new and it exists from 1992. DeVore, Jawerth and Lucier[10] presented a framework for analysing various algorithms for image compression based on wavelet approximations. Numerical implementation of fast wavelet packet image compression by adapting to a target image has been developed by Meyer, Averbuch and Stromberg [11]. Initially block based techniques such as DCT were used for image compression. A comparative study of 8x8 DCT and DWT has been performed and choice of an optimal wavelet function based on filter order, filter length, number of decomposition levels has been analysed by Grgic, Grgic, Cihlar[12]. Lawson and Zhu [13] have discussed the effective use and application of discrete wavelet transform compression algorithms in JPEG2000 image compression standard. Encoding scheme for Daubechies wavelets in image compression applications has been implemented by Wahid, Dimitrov, Julllien and Badawy[14]. Image compression using two dimensional DWT and estimating the detail matrices from the information matrix of the transformed image using adaptable RLS (Recursive Least Square) filter has been proposed by Sanchez, Meana, Miyatake [15]. A fully scalable image coding scheme based on SPIHT (Set Partitioning In Hierarchical Trees) algorithm was given by Danyali and Mertins [16] specifically for image transmission over networks and progressive web browsing. Chandler and Hemami[17] have proposed dynamic algorithm for maintaining visual quality using contrast based quantization in lossy wavelet image compression. A detailed study on image compression based on wavelet transform along with evaluation and comparison of seven wavelet families by applying them over test images has been done by Jain and Jain [18]. Chappelier and Guillemot [19] have designed oriented wavelet transform for image compression and denoising. Multi resolution image compression based on wavelet trees has been discussed by Oliver and Malumbres [20]. Kharate, Ghatol and Rege[21] have proposed image compression using wavelet packet tree decomposition. Shanon entropy is used to estimate the information content present in a component. If the sum of the information content present in the decomposed components is less than the original then decomposition is permitted and otherwise decomposition is stopped. Sudhakar, Karthiga, Jayaraman [22] conducted a survey on various methods of coding wavelet coefficients for image compression.

An improvement by decomposing the image into sub bands using analysis filter banks followed by quantization/coding and reconstruction of the full band image is introduced by Lin and Smith [23]. Spherical coding algorithm for wavelet image compression was designed and implemented by Ates and Orchard [24].

C. Hybrid K-Means and Wavelet Image Compression

Researches have been done involving K-means and wavelet image compression. The technique used by Venkateswaran and Rao [25] to achieve image compression is that instead of applying DWT over whole image, sub blocks of the image are subjected to wavelet decomposition and the wavelet coefficients are clustered using K-means clustering. Designing a vector quantization code book using fuzzy Probabilistic C Means clustering algorithm over wavelet packet tree coefficients has been implemented by Nagendran and Arockia Jansi Rani [26]. Krumm [27] has presented an image compression technique over multiple images using normalized cluster distance and nCut clustering using Partitioned iterated function systems. The study of noise over image compression using a neuro-fuzzy model based on wavelet transform has been presented by Vipula Singh [28].

II. PROPOSED WORK

During compression, a 256x256 gray scale image is represented as a two dimensional matrix. As the first step in encoding, the given image is subdivided into 4x4 blocks. Each block is then converted into one dimensional vector having 16 elements. The given image is now in the form of 4096 vectors and each with 16 elements. Now K-means clustering is applied over the 4096 vectors. Fig. 1 shows the steps involved in compression.

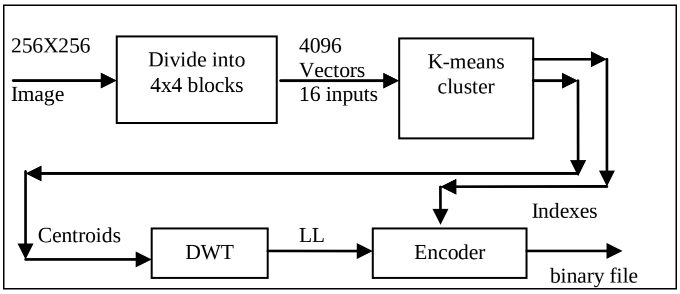

Figure. 1 Block diagram showing compression

The number of clusters is considered to be 256. Now 256 centroids or code vectors are created with 16 elements each. Each centroid is converted into 4x4 matrix and DWT is applied.

The primary idea is that after applying K-means clustering over the image vectors, the resulting 256 centroids are taken and DWT is applied over the centroids. Single level Haar transform is used since it has the advantages such as fast computation and memory efficiency. After applying wavelet transform to the 256 centroids, LH, HL and HH of the coefficient data can be removed from the resultant wavelet transform coefficients. The remaining LL coefficients can be encoded and stored along with the indexes. Fig. 2 shows single level Haar DWT decomposition.

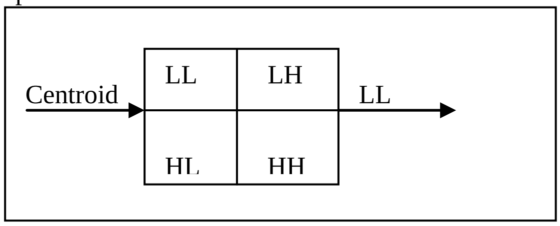

Figure. 2 Single level DWT

The input image is first thus divided into 4 x 4 blocks and converted into 4096 patterns of 16 bits length. Then the 4096 patterns corresponding to the 4096 blocks is taken and clustered by applying k- means clustering algorithm.

After clustering, there are 256 centroids each of size 16. DWT is applied over each centroid and four components are produced namely one approximation component and three detailed components. The detailed coefficients contain less important information and will not affect the visual quality of the image and hence in this work, only the approximation coefficients are stored. 4096 indexes corresponding to the input patterns of the 4096 image blocks generated by the clustering algorithm are stored. The LL component of each centroid has 4 elements. Since there are 256 centroids, 1024 elements have to be stored. Totally 5120 values are encoded into unsigned integer format requiring 1 byte each. This 5 KB information is then stored as a binary file. Thus the 256 x 256 image is now in the form of compressed binary file. The steps involved in the decompression process are shown in Fig. 3.

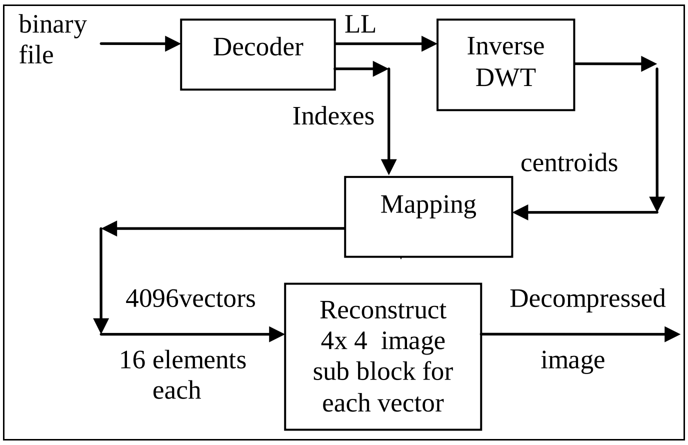

Figure. 3 Block diagram showing decompression

During decompression, the compressed binary file is taken as input. It is read and the indexes and LL components of the centroids are extracted and decoded into corresponding integer (indexes)/ floating point values(LL components). The LH, HL and HH components are assigned zeros. Inverse DWT is first applied over each 4x4 sub block formed using LL component keeping detailed components as zeros and 256 such centroids are reconstructed. Then retrieved 4096 indexes are mapped on to corresponding centroid from the reconstructed 256 centroids and 4096 output vectors each with 16 elements are restored. Then each output vector is converted into 4x4 sub block of the restored image.

III. IMPLEMENTATION AND EXPERIMENTAL RESULTS

During implementation of the proposed work, image compression using hybrid of K-means and discrete wavelet transform is performed. It is found that implementation of the proposed method gave better result. Many strategies are tested such as change in the number of clusters, change of size of input blocks, analysing and deciding the number of detailed coefficients to skip without storing them.

In most of the research papers it is found that first DWT is applied, the transformed coefficients are quantized and then only clustering techniques are applied to generate the centroids. The work on image compression using the wavelet transform and subsequent application of the clustering is a well-established concept. They focus on the selection of centroids for the clusters, which eventually form the codebooks. The indexing of the wavelet coefficients is done at the transmitter with respect to a code book using four clustering algorithms, namely K-means, fuzzy c-means, and genetic algorithm based clustering. The inverse process is performed at the receiver and the image is reconstructed. In this proposed technique the difference is that first K-means clustering is applied and then DWT is applied.

The proposed method is first applied over the test image Lena. The image is represented as a 2 dimensional matrix. After applying K-means and DWT, the indexes and LL components of the centroids are stored as binary file. The size of image lena.bmp is 65 KB. After compression the size of the compressed binary file of lena image is 5 KB. Since for any input image, 4096 indexes and LL components of 256 centroids are encoded and stored, the size of the compressed binary file for all 256x256 images is 5KB.

A. Algorithm for compression:

Step1: Split the image into Blocks of size NxN (4x4 or 8x8).
Step2: Rewrite each block as vector
Step3:Apply K-means algorithm over the centroids for clustering.
Step4: Apply DWT over each of the centroid.
Step5: Encode only the LL components and index entries.
Step6: Store the compressed image in binary file

B. Algorithm for decompression:

Step1: Read the indexes and LL components of centroids from the compressed file.
Step2: Apply inverse DWT and construct centroids.
Step3: Using indexes and centroids, reconstruct image block by block.

The proposed procedure is tested over other standard images also. Table I shows the file size of the original images and the compressed images. The size of the binary files obtained using the implementation of this method is compared with the size of the original file compressed using jpeg and is found that the size of the file obtained by this work is less than its jpeg equivalent. Thus better compression has been achieved. Table II shows the comparison between the compressed file size achieved by the proposed method and that of jpeg equivalent.

Table I Original File size and Compresses file size

Original Image	Original File Size	Size of compressed binary file
lena256.bmp	65KB	5 KB
girlface256.bmp	65 KB	5 KB
pepper256.bmp	65 KB	5 KB
zelda256.bmp	65 KB	5 KB
goldhill256.bmp	65 KB	5 KB
onion256.bmp	65 KB	5 KB

Table II Compressed File Size And JPEG File Size.

Original Image	JPEG File Size	Size of compressed binary file
lena256.bmp	9.37KB	5 KB
girlface256.bmp	9.48KB	5 KB
pepper256.bmp	9.73KB	5 KB
zelda256.bmp	9.32KB	5 KB
goldhill256.bmp	11.4KB	5 KB
onion256.bmp	7.49KB	5 KB

RMSE and PSNR between the original images and the reconstructed images are tabulated in Table III.

Table III RMSE and PSNR of reconstructed images.

Original Image	Reconstructed bmp file	RMSE	PSNR
lena256.bmp	lenareconstruct.bmp	4.8448	34.4252
girlface256.bmp	girlreconstruct.bmp	5.0446	34.0744
pepper256.bmp	pepperreconstruct.bmp	4.94	34.2563
zelda256.bmp	zeldareconstruct.bmp	4..8685	34.3829
goldhill256.bmp	goldhillreconstruct.bmp	6.1192	32.3969
onion256.bmp	onionsdreconstruct.bmp	4.1181	35.8369

Root Mean Square Error is given by the formula

$$\text{RMSE} = \sqrt{\frac{(\sum_{i=1}^{N}(x_i - \hat{x}_i)^2)}{N}}$$

Peak Signal To Noise Ratio is given by the formula

$$\text{PSNR} = 20\, log_{10} \frac{255}{RMSE}$$

Compression Ratio is given by the formula

$$\text{CR}=\frac{\text{Number of bits in the original image}}{\text{Number of bits in the compressed image}}$$

Table IV shows the Compression Ratio between original images and the compressed images.

Table IV Compression Ratio

Original Image	Reconstructed file	CR
lena256.bmp	lenareconstruct.bmp	13:1
girlface256.bmp	girlreconstruct.bmp	13:1
pepper256.bmp	pepperreconstruct.bmp	13:1
zelda256.bmp	zeldareconstruct.bmp	13:1
goldhill256.bmp	goldhillreconstruct.bmp	13:1
onion256.bmp	onionsdreconstruct.bmp	13:1

Analysis over the results achieved by earlier researches over 256 x 256 Lena image shows the following results. For Lena image, according to DeVore, Jawerth and Lucier[10], storing 20236 DWT coefficients required 14604 bytes and to store 12068 DWT coefficients required 8925 bytes. Our implementation requires only 5120 bytes and store 1024 coefficients and 4096 indexes. With hierarchical Partition Priority wavelet image compression, Efstratiadis, Tzovaras and Strintzis[3] achieved PSNR of 32dB using type I and type II filters at have 0.65 bpp. Grgic, Grgic, Cihlar[12] have got PSNR of 32.52dB at compression Ratio 10:1 whereas in the proposed work PSNR is 34.4252dB.

According to Lu and Shin [30], PSNR (Peak Signal to Noise Ratio) value obtained using LBG is 29.40dB and by the classified vector quantization technique of Lu and Shin PSNR is 30.03dB. Sanchez, Meana, Miyatake [15] obtained PSNR of 18.3135 at 0.5 bpp whereas our implementation results in 34.4252dB at 0.65bpp. With SPIHT Sudhakar, Karthiga, Jayaraman[22] made it 26.81dB at compression ratio 13.03 and 31.28dB at compression ratio of 6.57. Using curvelet, widgelet and ridgelet Joshi, Manthalkar and Joshi [35] achieved PSNR of 18.3dB, 17.57dB and 27.1dB over Lena image at compression factor of 7.49. When compared with all these results the proposed method outperforms them by achieving PSNR of 34.4252dB at compression factor of 7.69 and 0.65bpp. Laha, Pal and Chanda[31] in their work achieved PSNR of 33.03dB at compression rate 0.74bpp. Applying Linked Significant Tree method, Muzaffar and Choi [32] made PSNR of 24.58dB at 0.1dB, 27.22dB at 0.2bpp, 28.96dB at 0.3bpp, 30.81dB at 0.4bpp, 31.87dB at 0.5bpp. Chatellier, Boeglen, Perrine, Olivier and Haeberle [33] have got PSNR 31.34dB. When considering the PSNR values at various bpp resulting from all these techniques, our proposed work outperformed them by achieving PSNR of 34.4252dB at 0.625 bpp. Pandian and Anitha [34] used spatial quantization and obtained PSNR as 36.16db at compression Ratio 10.05:1. By designing K-means based image compression and wavelet based vector quantizer technique, Venkateswaran and Rao [25], have got PSNR 22.22dB and compression ratio 12.7. They applied dwt over the image followed by clustering. In our method, K-means clustering is performed which is followed by dwt and achieved PSNR value of 34.4252dB at compression ratio 13:1. Hence our work produces better compression. Table V shows the bit rate achieved by our implementation.

The results show that by this implementation good compression rate is achieved with acceptable loss in image quality and the reconstructed image maintains entropy level. Figure 4. Shows the original Lena image and reconstructed Lena image by our implementation.

Table V Bit Rate Of Compressed images.

Original image	Compressed binary file (.bin)	bits per pixel
lena256.bmp	lenakinter16.bin	0.625
girlface256	girlkinter16.bin	0.625
pepper256	pepperkinter16.bin	0.625
zelda256	zeldakinter16.bin	0.625
goldhill256	goldhillkinter16.bin	0.625
onion256	onionkinter16.bin	0.625

ORIGINAL IMAGE lena.bmp

RECONSTRUTED IMAGE lenareconstruct.bmp

Figure 4. Original image lena.bmp and reconstructed lena image lenareconstruct.bmp.

IV. CONCLUSION

A method for improving image compression using a hybrid of K-means and discrete wavelet transform is proposed and implemented successfully over 256 x 256 gray scale images. K-means is used as a vector quantizer and centroids are compressed by applying wavelet transform over each centroid. In this proposed method, a hybrid of K-means and DWT is selected. Both vector quantization using

K-means and wavelet transform are lossy compression techniques. This hybrid method leads to considerable reduction in the size of the file and the reconstructed images are visually acceptable. File size is considerably reduced because the compressed binary file contains the indexes and LL components of the centroids alone. The reconstruction is faster. From the experimental results of this technique it is obvious to note that this hybrid method leads to improved performance measures namely Peak Signal to Noise Ratio and bits per pixel over already existing techniques. The PSNR values are greater than 30 showing that the reconstructed images are acceptable [29]. The implementation works well with gray scale images of size 256 X 256.

V. REFERENCES

[1] Nicolaos B.Karayiannis, "Generalized fuzzy k-means algorithms and their application in image compression" in Proc. SPIE Applications of Fuzzy Logic Technology II, 1995, pp 206-217.

[2] Oleg Verevka and John W.Buchanan, "Local K-means Algorithm for Colour Image Quantization", in Proc. GI'95, Canadian Annual Conference, Quebec, Canada, 1995, pp 128-135.

[3] Serafim N.Efstratiadis, Dimitrios Tzovaras and Michael G.Strintzis, "Hierarchical Partition Priority Wavelet Image Compression", IEEE Trans. Image Processing, vol 5, No 7, pp 1111-1123, 1996.

[4] Daeryong Lee, Seongjoon Baek and Koengmo Sung, "Modified K-means Algorithm for Vector Quantizer Design", IEEE Signal Letters, vol 4, No.1, Jan 1997, pp 2-4.

[5] Tapas Kanungo, David M. Mount, Nathan S. Netanyahu, Christine D.Piatko, Ruth Silverman and Angela Y.Wu, "An Efficient k-Means Clustering Algorithm : Analysis and Implementation", IEEE Trans. Pattern Analysis and Machine Intelligence, vol 24, No.7, Jul 2002, pp 881-892.

[6] Dmitriy Fradkin, Ilya B.Muchnik, Simon Streltsov, Proc. International Conf. on Integration of Knowledge Intensive multi agent systems, 2003.

[7] C.H.Chou, M.C.Su, E.Lai, "A new cluster validity measure and its application in image compression", Pattern Analysis, vol 7, No.2, 2004, pp 205-220.

[8] Y.C. Hu and B.H.Su , "Accelerated k-means clustering algorithm for colour quantization", Imaging Science Journal, vol 56, No.1, Feb 2008, pp29-40.

[9] M.Emre Celebi, "Effective Initialization of K-means for color quantization", in Proc. ICIP09 of the 16th IEEE International Conference on Image Processing, 2009.

[10] Ronald A.DeVore, Bjorne Jawerth and Bradley J. Lucier, "Image Compression Through Wavelet Transform Coding", IEEE Trans. Information Theory, vol 38, No 2, pp 719-746, Mar 1992.

[11] Francois G.Meyer, Amir Z.Averbuch, and Jan-Olov Stromberg, "Fast Adaptive Wavelet Packet Image Compression", IEEE Trans. Image Processing, vol 9, No 5, pp 792-800, May 2000.

[12] Sonja Grgic, Mislav Grgic and Branka Zovko-Cihlar, "Performance Analysis of Image Compression Using Wavelets", IEEE Trans. Industrial Electronics, vol 48, No 3, pp 682-695, June 2001.

[13] S.Lawson and J.Zhu, "Image Compression using wavelets and JPEG2000: a tutorial", Electronics and Communication Engineering Journal, pp 112-121, June 2002.

[14] K.A. Wahid, V.S. Dimitrov, G.A. Julllien and W.Badawy, "Error-free computation of Daubechies wavelets for image compression applications", IEEE Electronics Letters, vol 39, No 5, pp 428-429, Mar 2003.

[15] Leornado E. Urbano Sanchez, Hector M. Perez Meana, Mariko Nakana Miyatake, "Wavelet Image Compression using universal coefficients matrix detail estimation" in Proc. 14th International Conference on Electronics, Communications and computers, IEEE, 2004.

[16] H.Danyali and A.Mertins, "Flexible, highly scalable, object based wavelet image compression algorithm for network applications" in IEE Proc. Image Signal Processing, 2004, pp 498-510.

[17] Damon M.Chandler, and Sheila S.Hemami, "Dynamic Contrast-Based Quantization for Losssy Wavelet Image Compression", IEEE Trans. Image Processing, vol 14, No 4, pp 397-410, 2005.

[18] Yogendra Kumar Jain and Sanjeev Jain, "Performance Evaluation of wavelets for Image Compression", Asian Journal of Information Technology, vol 5, No.10, 2006, pp 1104-1112.

[19] Vivien Chappelier and Christine Guillemot, "Oriented Wavelet Transform for Image Compressiona nd Denoising", IEEE Trans. Image P.rocessing, vol 15, No 10, 2892-2899, 2006.

[20] Jose Oliver and Manuel P.Malumbres, "Low-Complexity Multiresolution Image Compression Using Wavelet Lower Trees", IEEE Trans. Circuits and systems for video technology, vol 16, No 11, pp1437-1444, 2006.

[21] G. K. Kharate, A. A. Ghatol and P. P. Rege, "Image Compression Using Wavelet Packet Tree", GVIP Special Issue on Image Compression, ICGST, pp 25-28, 2007.

[22] R Sudhakar, R.Karthiga, S. Jayaraman, "Image Compression using Coding of Wavelet Coefficients - A Survey", GVIP Special Issue on Image Compression, ICGST, 2007.

[23] Jianyu Lin and Mark J. T. Smith, "New Perspectives and Improvements on the Symmetric Extension Filter Bank for Subband/Wavelet Image Compression", IEEE Trans. Image Processing, vol 17, No 2, pp 177-189, 2008.

[24] Hasan F.Ates and Michael T.Orchard, "Spherical Coding Algorithm for Wavelet Image Compression", IEEE Trans. Image Processing, vol 18, no 5, pp 1015-1024, 2009.

[25] N.Venkateswaran.and Y.V.Ramana Rao, "K-Means Clustering Based Image Compression in Wavelet Domain, Information Technology Journal, vol 6, No.1, 2007, pp 148-153.

[26] R.Nagendran and P.Arockia Jansi Rani, "A Novel FPCM based Vector Quantizer Codebook Design for Image Compression in the Wavelet Packet Domain", The International Journal of Multimedia and Its Applications, Vol2, No.1, Feb 2010, pp 1-7.

[27] Matthias Kramm, "Image Clusater Compression using Partitioned Iterated Function Systems and efficient Inter-Image Similarity Features", in SITIS'07 Pros of the Third International IEEE Conference on Signal-Image Technologies and Internet based System, 2007, China, pp 989-996.

[28] Vipula Singh, Navin Rajpal, K.Srikanta Murthy, "An Algorithmic Approach for efficient Image Compression using Neuro-Wavelet Model and Fuzzy Vector Quantization Technique", IJCSE International Journal on Computer Science and Engineering, vol 2,No.7, 2010, pp 2366-23.

[29] Ivan Vilovic, "An Experience in Image Compression Using Neural Networks", presented at the 48th International Symposium ELMAR, 2006, Zadar, Croatia, pp. 95-98.

[30] Cheng-Chang Lu and Yong Ho Shin, "A Neural Network based Image Compression System", Contributed paper, IEEE, 1992, pp 25-29.

[31] Arijit Laha, Nilhil R.Pal and Bhabatosh Chanda, "Design of Vector Quantizer for Image Compression Using Self Organizing Feature Map and Surface Fitting", IEEE Trans on Image Processing, vol 14, No.10, Oct 2004, pp 1291-1303.

[32] Tanzeem Muzaffar, Tae-Sun-Choi, "Linked Significant tree wavelet-based image compression", Signal Processing, vol 88, 2008, pp 2554-2563.

[33] C.Chatellier, H Boeglen, C,Perrine, C.Olivier O.Haeberle, “A robust joint source channel coding scheme for image transmission over the ionospheric channel”, Signal Processing: Image Communication, vol 22, 2007, pp 543-556.
[34] S.Immanuel Alex Pandian, J.Anitha, “A Neural Network Approach for Color image Compression in Transform Domain”, International Journal of Recent Trends in Engineering, vol 2, No.2, Nov 2009.
[35] M. S. Joshi, R. R. Manthalkar and Y. V. Joshi, “Image Compression Using Curvelet, Ridgelet and Wavelet Transform, A Comparative Study”, ICGST-GVIP, ISSN 1687-398X, 8(III), 2008, pp 25-34.

