Steganography Using Dictionary Sort on Vector Quantized Codebook

Dr. H. B. Kekre hbkekre@yahoo.com
MPSTME, NMIMS University, Vile-parle (W),
Mumbai-56, India

Archana Athawale athawalearchana@gmail.com
Ph.D. Scholar MPSTME, NMIMS University,
Mumbai-56, India Assistant Professor,
Thadomal Shahani Engineering College,
Bandra (W), Mumbai-50, India

Tanuja Sarode tanuja_0123@yahoo.com
Ph.D. Scholar MPSTME, NMIMS University,
Mumbai-56, India Assistant Professor,
Thadomal Shahani Engineering College,
Bandra (W), Mumbai-50, India

Sudeep Thepade sudeepthepade@gmail.com
Ph.D. Scholar MPSTME, NMIMS University,
Mumbai-56, India Assistant Professor,
MPSTME, NMIMS University,
Mumbai-56, India

Kalpana Sagvekar kalpanasagvekar@gmail.com
Lecturer, Fr. Conceicao Rodrigues
College of Engineering, Bandra (W),
Mumbai-50, India

Abstract

This paper presents a new reversible data hiding scheme using vector quantization (VQ). In traditional VQ based data hiding schemes secret data is hidden inside index based cover image resulting in limited embedding capacity. To improve the embedding capacity as well as to have minimum distortion to carrier media our method proves good. In this paper we have used four different codebook(CB) generation algorithms Linde Buzo and Gray (LBG), Kekre's Proportionate Error (KPE), Kekre's Median Codebook Generation algorithm (KMCG) and Kekre's Fast Codebook Generation Algorithm (KFCG) to prepare codebooks. So the Herculean task of increasing data hiding capacity with minimum distortion in recovered secret message is achieved with help of proposed techniques which are more robust against stegaanalysis.

Keywords: Data Hiding, VQ, LBG, KPE, KMCG, KFCG.

1. INTRODUCTION

The Internet has revolutionized the modern world and the numerous Internet based applications that get introduced these days add to the high levels of comfort and connectivity in every aspects of human life. As of September 2009, approximately 1.73 billion people worldwide use Internet for various purposes – ranging from accessing information for educational needs to financial transactions, procurement of goods and services [1]. As the modern world is gradually becoming "paperless' with huge amount of information stored and exchanged over the Internet, it is imperative to have robust security measurements to safeguard the privacy and security of the underlying data.

Cryptography techniques [2] have been widely used to encrypt the plaintext data, transfer the ciphertext over the Internet and decrypt the ciphertext to extract the plaintext at the receiver side. However, with the ciphertext not really making much sense when interpreted as it is, a hacker or an intruder can easily perceive that the information being sent on the channel has been encrypted and is not the plaintext. This can naturally raise the curiosity level of a malicious hacker or intruder to conduct cryptanalysis attacks on the ciphertext (i.e., analyze the ciphertext vis-à-vis the encryption algorithms and decrypt the ciphertext completely or partially) [2]. It would be rather more prudent if we can send the secret information, either in plaintext or ciphertext, by cleverly embedding it as part of a cover media (for example, an image, audio or video carrier file) in such a way that the hidden information cannot be easily perceived to exist for the unintended recipients of the cover media. This idea forms the basis for Steganography, which is the science of hiding information by embedding the hidden (secret) message within other, seemingly harmless images, audio, video files or any other media. Steganography protects the intellectual property rights and enables information transfer in a covert manner such that it does not draw the attention of the unintended recipients.

The existing schemes of data hiding can roughly be classified into the following three categories:
Spatial domain data hiding [3, 4, 5]: Data hiding of this type directly adjust image pixels in the spatial domain for data embedding. This technique is simple to implement, offering a relatively high hiding capacity. The quality of the stego image can be easily controlled. Therefore, data hiding of this type has become a well known method for image steganography.

Frequency domain data hiding [6, 7]: In this method images are first transformed into frequency domain, and then data is embedded by modifying the transformed coefficients.

Compressed domain data hiding [8, 9]: Data hiding is obtained by modifying the coefficients of the compressed code of a cover image. Since most images transmitted over Internet are in compressed format, embedding secret data into the compressed domain would provoke little suspicion.

In steganography the thirst is on increasing secret data hiding capacity and making the steganography techniques more and more robust against stegaanlysis attacks.

2. VQ COMPRESSION TECHNIQUE

Vector Quantization (VQ) [9-14] is an efficient technique for data compression [31-34] and is very popular in a variety of research fields such as data hiding techniques [7,8], image segmentation [23-26], speech data compression [27], content based image retrieval CBIR [28, 29] and face recognition [30].

2.1 Codebook Generation Algorithms

2.1.1. Linde-Buzo-Gray (LBG) Algorithm [10]

In this algorithm centroid is calculated as the first codevector for the training set. In Figure 1 two vectors v1 & v2 are generated by using constant error addition to the codevector. Euclidean distances of all the training vectors are computed with vectors v1 & v2 and two clusters are formed based on nearest of v1 or v2. This procedure is repeated for every cluster. The drawback of this algorithm is that the cluster elongation is -45^{o} to horizontal axis in two dimensional cases. Resulting in inefficient clustering.

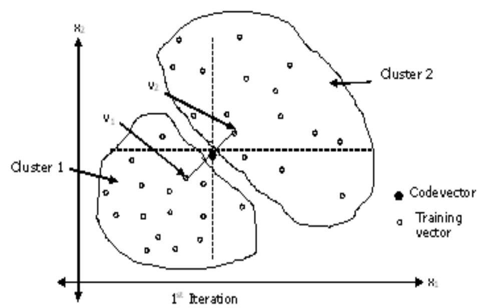


FIGURE 1: LBG for 2 dimensional case

2.1.2. Proportionate Error Algorithm (KPE) [11], [12]

Here proportionate error is added to the centroid to generate two vectors v1 & v2. Magnitude of elements of the centroid decides the error ratio. Hereafter the procedure is same as that of LBG. While adding proportionate error a safe guard is also introduced so that neither v1 nor v2 go beyond the training vector space. This removes the disadvantage of the LBG. Both LBG and KPE requires 2M number of Euclidean distance computations and 2M number of comparisons where M is the total number of training vectors in every iteration to generate clusters.

2.1.3. Kekre's Median Codebook Generation Algorithm (KMCG) [13]

In this algorithm image is divided into blocks and blocks are converted to the vectors of size k. The Figure 2 below represents matrix T of size M x k consisting of M number of image training vectors of dimension k. Each row of the matrix is the image training vector of dimension k.

$$T = \begin{bmatrix} x_{1,1} & x_{1,2} & \cdots & x_{1,k} \\ x_{2,1} & x_{2,2} & \cdots & x_{2,k} \\ . & . & . & . \\ . & . & . & . \\ x_{M,1} & x_{M,2} & \cdots & x_{M,k} \end{bmatrix}$$

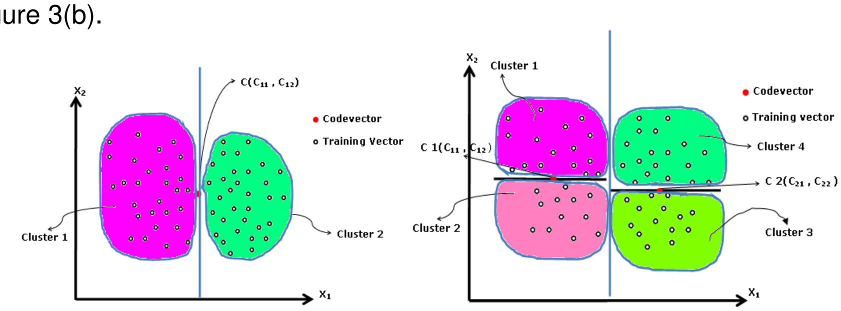
FIGURE 2: Training Matrix

The training vectors are sorted with respect to the first member of all the vectors i.e. with respect to the first column of the matrix T and the entire matrix is considered as one single cluster. The median of the matrix T is chosen (codevector) and is put into the codebook, and the size of the codebook is set to one. The matrix is then divided into two equal parts and the each of the part is then again sorted with respect to the second member of all the training vectors i.e. with respect to the second column of the matrix T and we obtain two clusters both consisting of equal number of training vectors. The median of both the parts is the picked up and written to the codebook, now the size of the codebook is increased to two consisting of two codevectors and again each part is further divided to half. Each of the above four parts obtained are sorted with respect to the third column of the matrix T and four clusters are obtained and accordingly four codevectors are obtained. The above process is repeated till we obtain the codebook of desired size. Here quick sort algorithm is used and from the results it is observed that this algorithm takes least time to generate codebook, since Euclidean distance computation is not required.

2.1.4. Kekre's Fast Codebook Generation (KFCG) Algorithm[14]

In [14], KFCG algorithm for image data compression is proposed. This algorithm reduces the time for codebook generation. It does not use Euclidian distance for codebook generation. In this algorithm image is divided into blocks and blocks are converted to the vectors of size k. Initially we have one cluster with the entire training vectors and the codevector C_1 which is centroid.
In the first iteration of the algorithm, the clusters are formed by comparing first element of training vector with first element of codevector C_1. The vector Xi is grouped into the cluster 1 if $x_{i1} < c_{11}$ otherwise vector X_i is grouped into cluster 2 as shown in Figure 3(a). where codevector dimension space is 2.

In second iteration, the cluster 1 is split into two by comparing second element x_{i2} of vector Xi belonging to cluster 1 with that of the second element of the codevector which is centroid of cluster 1. Cluster 2 is split into two by comparing the second element x_{i2} of vector X_i belonging to cluster 2 with that of the second element of the codevector which is centroid of cluster 2, as shown in Figure 3(b).



3(a). First Iteration 3(b). Second Iteration

FIGURE 3: KFCG algorithm for 2-D case

This procedure is repeated till the codebook size is reached to the size specified by user. It is observed that this algorithm gives less error as compared to LBG and requires least time to generate codebook as compared to other algorithms, as it does not require computation of Euclidian distance.

3. PROPOSED APPROACH

In this approach, we are hiding the secret data into codebook generated using various codebook generation algorithm such as LBG[10], KPE[12][13], KMCG[14], KFCG[15]. There are various ways of hiding: 1bit, 2 bits, 3 bits, 4 bits & variable bits hiding.

The algorithm is as follows:

a. Divide the image into 2×2 block of pixels.
b. Generate initial cluster of training set using the rows of 12 values per pixel window.
c. Apply codebook generation algorithm LBG/KPE/KFCG/KMCG on initial cluster to obtain codebook of size 2048 codevectors.
d. Add initial index position column in codebook (CB).
e. Perform dictionary sort on CB.
f. Hide data into sorted CB except the last column.
g. Again add final index position column in Stego CB.
h. Sort the stego CB.
i. From sorted CB reconstruct the image.
j. Send the reconstructed image & new final index position to receiver.
k. Receiver will divide image into blocks generating training vectors. Collection of unique training vector is nothing but CB.
l. Arrange the entries of codebook using final index position
m. Extract the secret data.

3.1 Variable Bit Hiding Algorithm

For variable bit hiding Kekre's algorithm [3] is used.

a. If the value of codebook vector element is in the range 240 ≤ gi ≤ 255 then we embed 4 bits of secret data into the 4 LSB's codebook vector element. This can be done by observing the 4 most significant bits (MSB's). If they are all 1's then the remaining 4 LSB's can be used for embedding data.
b. If the value of codebook vector element is in the range 224 ≤ gi ≤ 239 then we embed 3 bits of secret data. . This can be done by observing the 3 most significant bits (MSB's). If they are all 1's then the remaining 3 LSB's can be used for embedding data.
c. If the value of codebook vector element is in the range 192 ≤ gi ≤ 223 then we embed 2 bits of secret data. . This can be done by observing the 2 most significant bits (MSB's). If they are all 1's then the remaining 2 LSB's can be used for embedding data.
d. If the value of codebook vector element is in the range 0 ≤ gi ≤ 191 we embed 1 bit of secret data.

4. RESULTS AND EVALUATION

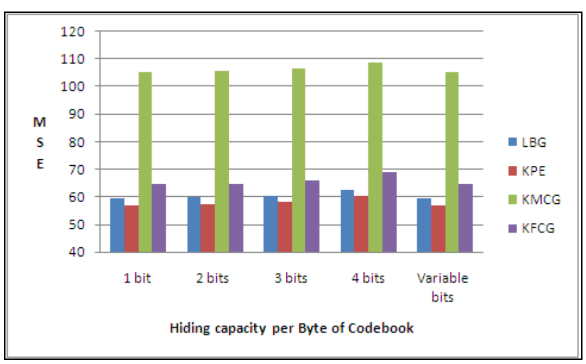
In our proposed approach, we have generated codebook using LBG, KPE, KMCG and KFCG for 24 bit color image of size 256×256 shown in Figure 4(a) to 8(a). Codebook is of size 2048×12 (i.e. 2048 code vectors each contains 12 bytes - 4 pairs of RGB). We have hidden 32×32 gray image.

Figure 4 to Figure 8 Shows the results of 1 bit, 2 bits, 3 bits, 4 bits and Variable bits using codebook obtained from LBG, KPE, KMCG and KFCG on the various cover images Eleafrica, Bird, Panda, Flowers and Manydogs hiding same secrete image for fair comparison respectively.

4 (a) Original Cover image Eleafrica.bmp

4 (b) Secret Message

4 (c) Reconstructed image using KPE for Variable bits Method



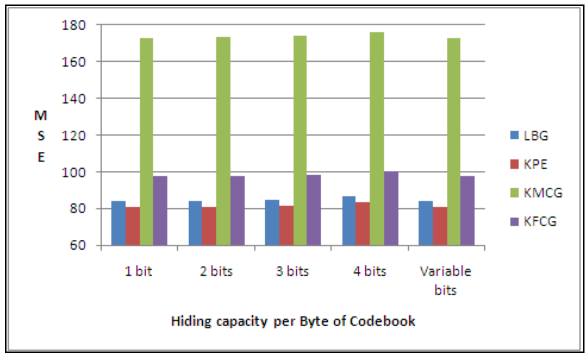
4 (d) Plot of MSE versus Hiding Capacity

FIGURE 4: Results of 1bit, 2bits, 3 bits, 4bits and Variable bits on the cover image Eleafrica and secrete image shown in Figure 4(b).

5 (a)
Original
Cover image
Bird.bmp

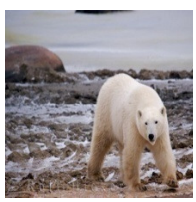
5 (b)
Secret
Message

5(c)
Reconstructed image
using KPE for
Variable bits Method



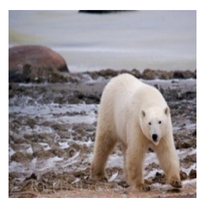
5 (d) Plot of MSE versus Hiding Capacity

FIGURE 5: Results of 1bit, 2bits, 3bits, 4bits and Variable bits on the cover image Bird and secrete image shown in Figure 5(b).

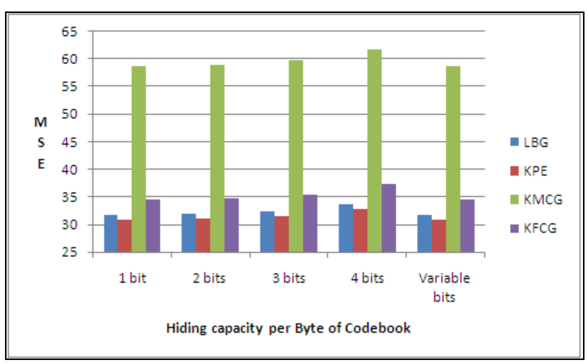


6 (a)
Original
Cover image
Panda.bmp

6 (b)
Secret
Message



6 (c)
Reconstructed image
using KPE for
Variable bits Method



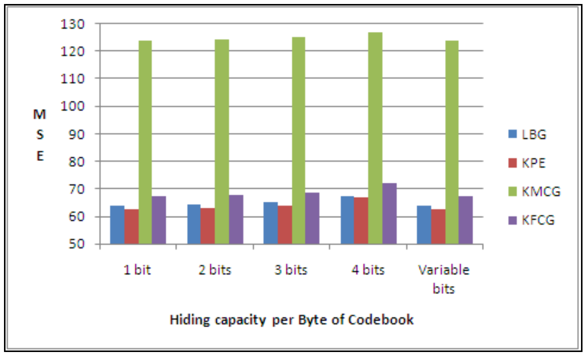
6 (d) Plot of MSE versus Hiding Capacity]

FIGURE 6: Results of 1bit, 2bits, 3bits, 4bits and Variable bits on the cover image Panda and secrete image shown in Figure 6(b).

7 (a) Original Cover image Flowers.bmp

7 (b) Secret Message

7 (c) Reconstructed image using KPE for Variable bits Method



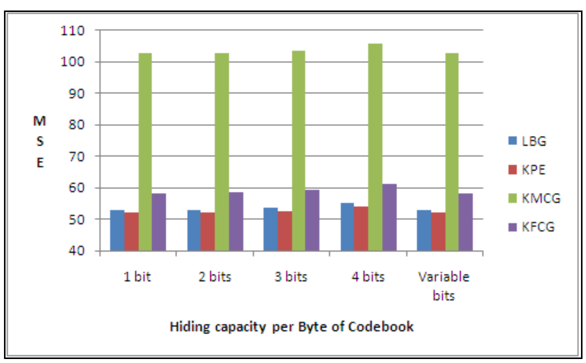
7 (d) Plot of MSE versus Hiding Capacity

FIGURE 7: Results of 1bit, 2bits, 3bits, 4bits and Variable bits on the cover image Flowers and secrete image shown in Figure 7(b).

8 (a) Original Cover image Manydogs.bmp

8 (b) Secret Message

8(c) Reconstructed image using KPE for Variable bits Method

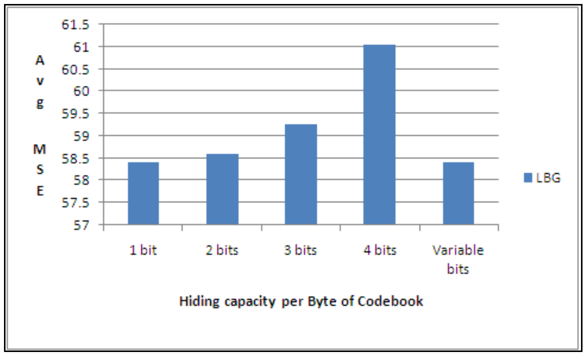


8 (d) Plot of MSE versus Hiding Capacity

FIGURE. 8: Results of 1bit, 2bits, 3bits , 4bits and Variable bits on the cover image Manydogs and secrete image shown in Fig. 8(b).

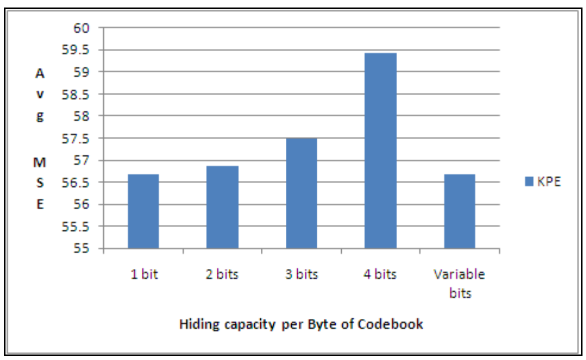
It is observed from Figure 4 to Figure 8 that KPE algorithm gives less MSE in all the data hiding methods 1bit, 2bits, 3bits, 4bits and variable bits as compared to LBG, KMCG and KFCG codebook. LBG and KPE results are comparable.

Figure 9 to 12 shows the results of avg mse versus hiding capacity for various codebook generation techniques by taking average of MSEs for 1 bit, 2 bits, 3 bits, 4 bits and variable bits hiding methods on the various cover images Eleafrica, Bird, Panda, Flowers and Manydogs hiding same secrete image for fair comparison respectively.



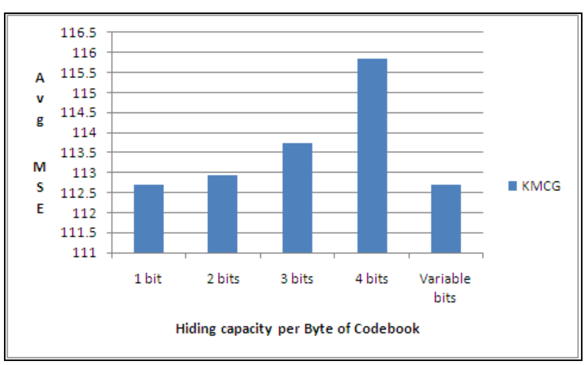
Plot of Avg. MSE versus Hiding Capacity

FIGURE 9: Plot of Hiding Capacity versus average MSE for various hiding methods 1bit, 2bits, 3bits, 4bits and Variable bits on LBG.



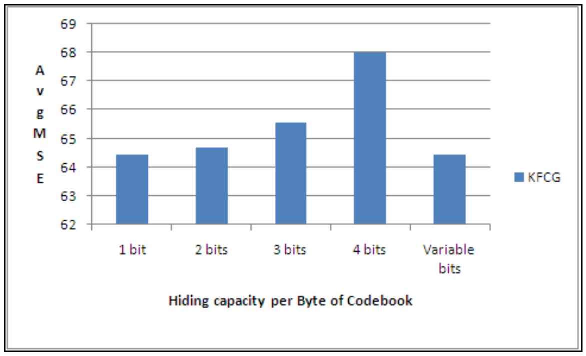
Plot of Avg. MSE versus Hiding Capacity

FIGURE 10: Plot of Hiding Capacity versus average MSE for various hiding methods 1bit, 2bits,3bits, 4bits and Variable bits on KPE.



Plot of Avg. MSE versus Hiding Capacity

FIGURE 11: Plot of Hiding Capacity versus average MSE for various hiding methods 1bit, 2bits,3bits, 4bits and Variable bits on KMCG.



Plot of Avg. MSE versus Hiding Capacity

FIGURE 12: Plot of Hiding Capacity versus average MSE for various hiding methods 1bit, 2bits,3bits, 4bits and Variable bits on KFCG.

From Fig. 9 to Fig 12 it has been observed that variable bit hiding method gives better embedding capacity coupled with low distortion.

5. CONCLUSION

Cover image size always dependant on secret image size to be embedded but by using the proposed technique cover image size can be made independent of secret image size. As well as in our proposed technique Secret message get spread throughout the image therefore conventional stegaanalysis technique fail to detect secret data. It is been observed that LBG and KPE gives less MSE compared to other algorithms, proving to be better for steganographic application than KFCG & KMCG. But KFCG and KMCG are faster than KPE and LBG. So tremendous amount of time is saved in the whole process because of using KFCG/KMCG at the cost of slightly more distortion in retrieved messages. Variable bit hiding gives lowest distortion in all VQ codebook generation techniques with considerably large secret data hiding capacity as compared to 1 bit, 2 bits, 3 bits and 4 bits hiding techniques.

6. REFERENCES

[1] Available at: http://www.internetworldstats.com/stats.htm [Accessed 7 July 2010]

[2] D. Stinson. *"Cryptography: Theory and Practice",* 2nd Edition, Chapman and Hall/ CRC, February 2002

[3] H. B. Kekre, A. Athawale and P. N.Halarnkar. *"Increased Capacity of Information Hiding in LSBs Method for Text and Image".* International Journal of Electrical, Computer and Systems Engineering, 2(4) Available at: http://www.waset.org/ijecse/v2.html.

[4] H. B. Kekre, A. Athawale and P. N. Halarnkar. *"Polynomial Transformation To Improve Capacity Of Cover Image For Information Hiding In Multiple LSBs".* International Journal of Engineering Research and Industrial Applications (IJERIA), Ascent Publications, Pune, II: 2009

[5] H. B. Kekre, A. Athawale and P. N. Halarnkar. *"Performance Evaluation Of Pixel Value Differencing And Kekre's Modified Algorithm For Information Hiding In Images".* ACM International Conference on Advances in Computing, Communication and Control (ICAC3).2009 Available at: http://portal.acm.org/citation.cfm?id=1523103.1523172.

[6] S. D. Lin, C. F. Chen. *"A Robust DCT-based Watermarking for Copyright Protection".* IEEE Transactions on Consumer Electron, 46(3):415-421, 2000

[7] Y.T. Wu, F.Y. Shih. *"Genetic algorithm based methodology for breaking the steganalytic systems"*. IEEE Transactions on Systems, Man and Cybernetics. Part B, 36(1):24-31, 2006

[8] C. C. Chang, C. Y. Lin. *"Reversible Steganography for VQ-compressed Images Using Side Matching and Relocation"*. IEEE Transactions on Information Forensics and Security, 1(4): 493-501, 2006

[9] C. C. Chang, Y. C. Chou, C. Y. Lin. *"Reversible Data Hiding in the VQ-Compressed Domain"*. IEICE Transactions on Information and Systems, E90-D(9):1422-1429, 2007

[10] A. Gersho, R. M. Gray. *"Vector Quantization and Signal Compressio"*, Kluwer Academic Publishers, Boston, MA, (1991)

[11] H. B. Kekre, T. K. Sarode. *"New Fast Improved Codebook generation Algorithm for Color Images using Vector Quantization"*. International Journal of Engineering and Technology, 1(1):67-77, 2008

[12] H. B. Kekre, T. K. Sarode. *"An Efficient Fast Algorithm to Generate Codebook for Vector Quantization"*. First International Conference on Emerging Trends in Engineering and Technology, ICETET-2008, held at Raisoni College of Engineering, Nagpur, India, 2008 Avaliable at: online IEEE Xplore

[13] H. B. Kekre, T. K. Sarode. *"Fast Codebook Generation Algorithm for Color Images using Vector Quantization"*. International Journal of Computer Science and Information Technology, 1(1):7-12, 2009

[14] H. B. Kekre, T. K. Sarode. *"New Fast Improved Codebook Generation Algorithm for Color Images using Vector Quantization"*. International Journal of Engg. & Tech., 1(1):67-77, 2008

[15] R. M. Gray. *"Vector quantization"*. IEEE Acoust., Speech, Signal Process., 1:4–29, 1984

[16] T. Kim. *"Side match and overlap match vector quantizers for images"*. IEEE Trans. Image Process., 1(4):170–185,1992

[17] W. B. Pennebaker and J. L. Mitchell. "The JPEG Still Image Data Compression Standard". New York: Reinhold, 1993

[18] D. S. Taubman and M. W. Marcellin. *"JPEG2000: Image Compression Fundamentals Standards and Practice"*. Norwell, MA: Kluwer, 2002

[19] A. Gersho and R. M. Gray. *"Vector Quantization and Signal Compression"*. Norwell, MA: Kluwer, 1992

[20] Z. N. Li and M. S. Drew. "Fundamentals of Multimedia. Englewood Cliffs". NJ: Prentice-Hall, Oct. 2003

[21] N. M. Nasrabadi, R. King. *"Image coding using vector quantization: A review"*. IEEE Trans. Commun., 36(8):957–971, 1988

[22] C. H. LEE, L. H. CHEN. *"Fast Codeword Search Algorithm for Vector Quantization"*. IEE Proceedings Image Signal Processing 141(3): 1994

[23] H. B. Kekre, T. K. Sarode, B. Raul. *"Color Image Segmentation using Kekre's Fast Codebook Generation Algorithm Based on Energy Ordering Concept"*. ACM International Conference on Advances in Computing, Communication and Control (ICAC3-2009), pp.:

357-362, 2009. Fr. Conceicao Rodrigous College of Engg., Mumbai. Available at: ACM portal.

[24] H. B. Kekre, T. K. Sarode, B. Raul. *"Color Image Segmentation using Kekre's Algorithm for Vector Quantization"*. International Journal of Computer Science (IJCS), 3(4): 287-292, 2008. Available at: http://www.waset.org/ijcs.

[25] H. B. Kekre, T. K. Sarode, B. Raul. *"Color Image Segmentation using Vector Quantization Techniques Based on Energy Ordering Concept"*. International Journal of Computing Science and Communication Technologies (IJCSCT) 1(2):164-171, 2009

[26] H. B. Kekre, T. K. Sarode, B. Raul. *"Color Image Segmentation Using Vector Quantization Techniques"*. Advances in Engineering Science Sect. C (3): 35-42, 2008

[27] H. B. Kekre, T. K. Sarode. *"Speech Data Compression using Vector Quantization"*. WASET International Journal of Computer and Information Science and Engineering (IJCISE), 2(4):251-254, 2008 Available at: http://www.waset.org/ijcise.

[28] H. B. Kekre, Ms. T. K. Sarode, S. D. Thepade. *"Image Retrieval using Color-Texture Features from DCT on VQ Codevectors obtained by Kekre's Fast Codebook Generation"*. ICGST-International Journal on Graphics, Vision and Image Processing (GVIP), 9(5):1-8, 2009. Available at: http://www.icgst.com/gvip/Volume9/Issue5/P1150921752.html.

[29] H. B. Kekre, T. Sarode, S. D. Thepade. *"Color-Texture Feature based Image Retrieval using DCT applied on Kekre's Median Codebook"*. International Journal on Imaging (IJI), 2(A09):55-65, 2009. Available at: www.ceser.res.in/iji.html (ISSN: 0974-0627).

[30] H. B. Kekre, K. Shah, T. K. Sarode, S. D. Thepade. *"Performance Comparison of Vector Quantization Technique – KFCG with LBG, Existing Transforms and PCA for Face Recognition"*. International Journal of Information Retrieval (IJIR), 02(1):64-71, 2009

[31] H. B. Kekre, T. K. Sarode. *"2-level Vector Quantization Method for Codebook Design using Kekre's Median Codebook Generation Algorithm"*. Advances in Computational Sciences and Technology (ACST), ISSN 0973-6107, 2(2):167–178, 2009. Available at: http://www.ripublication.com/Volume/acstv2n2.htm.

[32] H. B. Kekre, T. K. Sarode. *"Multilevel Vector Quantization Method for Codebook Generation"*. International Journal of Engineering Research and Industrial Applications (IJERIA), 2(V):217-235, 2009, ISSN 0974-1518. Available at: http://www.ascent-journals.com/ijeria_contents_Vol2No5.htm.

[33] H. B. Kekre, T. K. Sarode. *"Vector Quantized Codebook Optimization using K-Means"*. International Journal on Computer Science and Engineering (IJCSE) 1(3):283-290, 2009 Available at: http://journals.indexcopernicus.com/abstracted.php?level=4&id_issue=839392

[34] H. B. Kekre, T. K. Sarode. *"Bi-level Vector Quantization Method for Codebook Generation"*. Second International Conference on Emerging Trends in Engineering and Technlogy, at G. H. Raisoni College of Engineering, Nagpur, 2009, this paper will be uploaded online at IEEE Xplore