Performance Comparison of LBG, KPE, KFCG and KMCG for Global Codebook Technique

Dr. H. B. Kekre
Sr. Professor
MPSTME, NMIMS
University, Vileparle(w)
Mumbai -56,
India.

Dr.Tanuja K. Sarode
Asst. Professor,
Thadomal Shahni
Engineering College,
Bandra(W),
Mumbai – 50, India

Ms.Kavita Raut
Lecturer
Thadomal Shahani
Engineering College,
Bandra(W),
Mumbai – 50, India.

Rohan Tahiliani
MS Information Security
Georgia Institute of
Technology
Atlanta 30332, GA,
USA.

ABSTRACT
Vector quantization is a classical quantization technique from signal processing which allows the modeling of probability density functions by the distribution of prototype vectors. It was originally used for data compression. It works by dividing a large set of points (vectors) into groups having approximately the same number of points closest to them. Each group is represented by its centroid point, as in clustering algorithms.

Vector Quantization is a technique of compressing data based on grouping blocks having similar data. These blocks are called Code Vectors and all the code vectors grouped together is called a Codebook. The key to VQ data compression is a good codebook. In order to reduce bandwidth overhead it is necessary to generate Global Codebook for a particular class of images. Otherwise local codebook has to be transferred every time before the transmission of image. In this paper various global codebook generation algorithms for vector quantization for color images are presented.

Keywords
Vector Quantization, Clustering, Global Codebook.

1. INTRODUCTION

1.1 Vector Quantization:-
Images are used for a communication from ancient age and because of the rapid technological growth and the usage of the internet today storage and transmission of digital data/image is possible today. Also the transmission of multimedia applications over the web is increasing day by day. The multimedia applications consist of mainly speech, images, and videos. These applications requires large amount of data resulting in consumption of huge bandwidth and storage resources.

Vector quantization (VQ) [1]-[3] is an efficient technique for data compression and has been successfully used in various applications involving VQ-based encoding and VQ based recognition. The response time is very important factor for real time application [1]. Many type of VQ, such as classified VQ [9], [10], address VQ [9], [11], finite state VQ[9], [12], side match VQ[9], [13], mean-removed classified VQ[9], [14], and predictive classified VQ[9], [15], have been used for various purposes

Vector Quantization (VQ) is a compression technique based on grouping blocks of information based on the similarity of their values. There is a loss of quality while using VQ, but this is duly compensated by the significant savings achieved by this compression method. VQ leads to formation of Codebooks. These Codebooks are a subset of the blocks derived from the data. It is an iterative method of clustering data, where iteration involves increasing the number of clusters twofold and re-clustering the data till a finite desired number of clusters is reached. It is a three phase process involving Codebook Generation, Encoding and Decoding.

The density matching property of vector quantization is powerful, especially for identifying the density of large and high-dimensioned data. Since data points are represented by the index of their closest centroid, commonly occurring data have low error, and rare data high error. This is why VQ is suitable for lossy data compression. It can also be used for lossy data correction and density estimation [16].

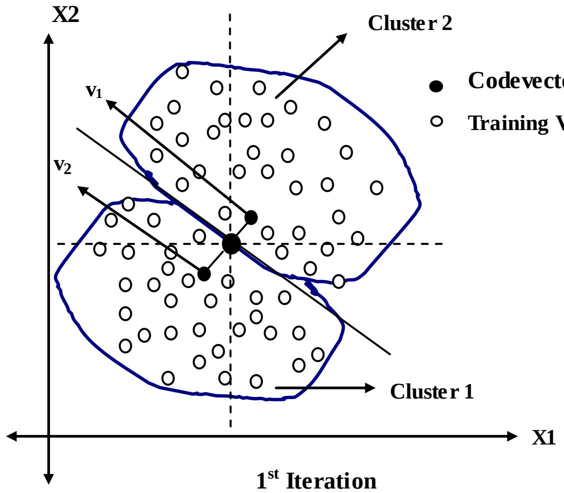
Vector Quantization (VQ) [17-25] is an efficient technique for data compression and has been successfully used in various applications such as index compression [26, 27]. VQ has been very popular in a variety of research fields such as speech recognition and face detection [28, 29]. VQ is also used in real time applications such as real time video-based event detection and anomaly intrusion detection systems [30], image segmentation [31-32], speech data compression [33], content based image retrieval CBIR [34, 35] and face recognition [36].

Codebook can be generated using clustering algorithms in spatial or in transform domain. Codebook is present at the both ends of the communication. Hence before encoding of the image, codebook should be sent to receiver's end. In order to speed up the transmission of images it is necessary to construct the global codebook for a particular class of images. Otherwise local codebook has to be transferred every time before the transmission of the image. This will be bandwidth overhead.

2. ALGORITHMS FOR CODEBOOK GENERATION
In this section LBG, KPE, and KFCG Codebook generation method are discussed.

2.1 Linde-Buzo-Gray (LBG) Algorithm [16]:-

In this algorithm centroid is computed as the first codevector for the training set. Two vectors v1 & v2 are then generated by adding constant error to the codevector as shown in Figure 1. 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.



1st Iteration
Figure 1. LBG algorithm for 2-D case

2.2 Kekre's Proportionate Error (KPE) Algorithm[17,18]:-

In this algorithm a proportionate error is added to the centroid to generate two vectors v1 & v2. The error ratio is decided by magnitude of coordinates of the centroid. Hereafter the procedure is same as that of LBG.

2.3 Kekre's Fast Codebook Generation (KFCG) Algorithm [20,21]:-

This algorithm reduces the time for codebook generation. It does not use Euclidian distance for codebook generation. In this algorithm image is divided in to blocks and blocks are converted to the vectors of size k. Initially only one cluster with the entire training vectors and the codevector C1 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 code vector C1. The vector X_i is grouped into the cluster 1 if $x_{i1} < c_{11}$ otherwise vector X_i is grouped into cluster 2 as shown in Figure 2a. where code vector dimension space is 2.

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

This procedure is repeated till the codebook size is reached to the size specified by user.

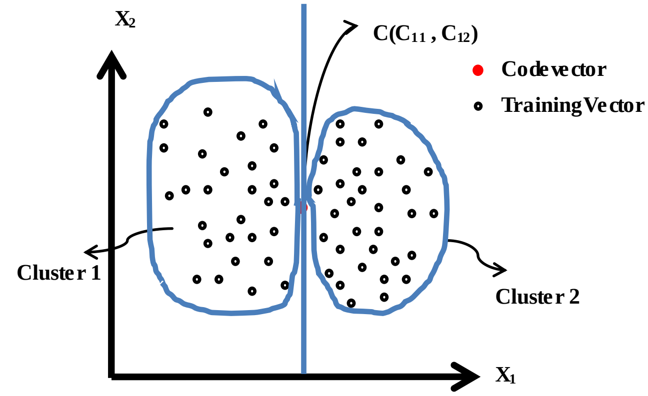


Figure 2a. 1st Iteration

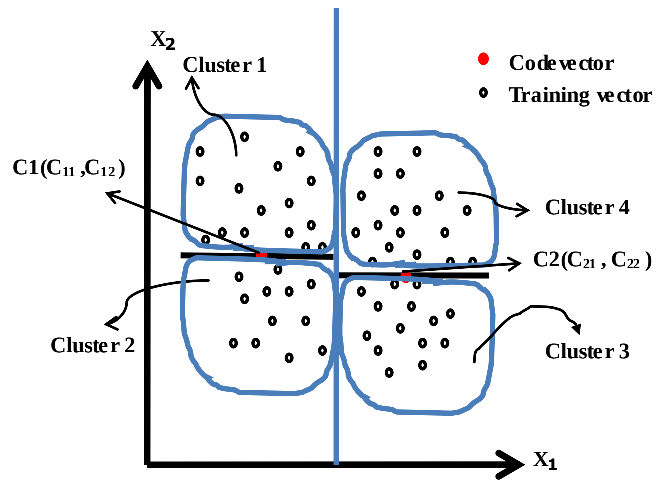


Figure 2b. 2nd Iteration

Figure 2. KFCG algorithm for 2-D case

2.4 Kekre's Median Codebook Generation (KMCG) Algorithm [19]:-

In this algorithm image is divided in to blocks and blocks are converted to the vectors of size k. The Fig. 3 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.

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

Figure 3: Matrix of image training vectors

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 (code vector) 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 code vectors 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 code vectors are obtained. The above process is repeated till we obtain the codebook of desired size. Here quick sort algorithm is used.

3. PROPOSED GLOBAL CODEBOOK METHOD:

In this paper two Categories are selected: car and facial category. Each category consists of 10 colored bit map images of size 256x256x3. Global codebook is created from the 10 local codebooks.

Steps for the algorithm:-

1. For each image belonging to similar category, divide the image into 2x2 non overlapping blocks of pixels and each image block is converted to vector of dimension 12 Convert each block into training vector.
2. Generate Local Codebook of desired sizes 128x12, 256x12, 512x12 and 1024x12 using LBG/KPE/ KFCG/KMCG respectively for each image belonging to the same class.
3. Merge all 10 local codebooks LBG/KPE/KFCG/KMCG of sizes 128x12 to 1024x12 respectively to one and create four merged large codebooks of size 1280x12, 2560x12, 5120x12 and 10240x12 respectively.
4. On each of these large codebooks apply LBG/KPE/KFCG/KMCG and generate global codebook of sizes 128x12 to 1024x12.

4. RESULTS AND DISSCUSSIONS

The Global Codebook Technique using two sets of data for each category are implemented. For each category Training Set containing 10 color 256x256x12 bitmap images as shown in Figure 4 and Figure 5, and five color images of size 256x256x12 as Test Set as shown in Figure 6 and Figure 7 respectively are used. The Training set is used to generate the Global Codebook and Test set is used to measure how effectively the Global Codebook can compress the images.

The block size used while encoding and decoding the images is 2x2 pixels. All the three mentioned algorithms on both the categories for Codebook Sizes of 128, 256, 512 and 1024 are tested. The Global Codebook was always generated by the same algorithm used to generate the individual codebooks. To measure the distortion after compression the Mean Squared Error (MSE) between the Result Image and the Original Image are calculated. The images are numbered in each category starting from 1, going left to right and top to bottom.

Figure 4. Shows the 10 Training Images of category Car. Figure 5. Shows the 10 Training Images of category Facial. Figure 6. shows the 5 Test Images of category Car. Figure 7. Shows the 5 Test Images of category Facial.

Figure 4. Training Images of category Car

Figure 5. Training Images of category Facial

Figure 6. Test Images of category Car

Figure 7. Test Images of category Facial

Table 1. Shows the MSE for all the 10 images in Training set for car category using all four algorithms for different Codebook sizes 128, 256, 512 and 1024. Figure 8. Shows the Average MSE obtained using all four clustering algorithms with Codebook sizes 128, 256, 512 and 1024 for car category. It is observed that for all the codebook sizes 128 to 1024 the MSE obtained from the KFCG is less as compared to LBG ,KMCG and KPE.

Table 1. MSE for the images in Training set for car category using all four algorithms for different Codebook sizes 128, 256, 512 and 1024.

Image No.	Codebook size 128				Codebook size 256				Codebook size 512				Codebook size 1024			
	LBG	KPE	KFCG	KMCG	LBG	KPE	KFCG	KMCG	LBG	KPE	KFCG	KMCG	LBG	KPE	KFCG	KMCG
1	914.03	635.72	195.79	741.85	939.33	305.15	146.38	204.48	883.39	218.48	115.89	149.14	858.41	144.56	82.644	112.28
2	336.48	304.51	152.43	335.9	322.99	267.7	117.86	190.21	307.75	177.18	88.044	118.33	284.38	107.12	58.514	83.717
3	356.14	343.4	189.73	279.96	346.64	311.37	131.11	221.31	329.67	205.72	79.797	126.57	305.97	112.61	48.54	82.554
4	249.18	199.42	122.44	243.53	250	156.51	87.012	139.26	228.46	107.26	62.916	86.265	211.9	79.916	46.377	64.257
5	264.91	257.66	124.85	306.09	259.88	234.76	81.987	161.63	248.81	150.01	58.329	93.904	224.4	79.702	30.608	60.243
6	868.38	257.66	283.88	898.24	883.55	427.29	223.56	337.16	820.79	297.6	183.85	242.43	803.18	214.37	128.23	186.8
7	180.57	168.48	106.24	192.32	175.44	150.52	62.86	124.67	163.26	90.503	47.597	70.035	141.85	55.875	34.179	52.735
8	662.66	357.91	193.87	666.32	609.13	298.63	139.54	211.68	623.35	207.99	109.08	141.73	562.45	135.69	70.793	103.62
9	277.84	249.76	162.77	315.87	273.19	230.95	117.7	189.3	255.93	147.56	78.209	110	233.61	98.065	53.495	76.995
10	224.54	220.18	118.15	240.68	218.85	200.82	78.629	137.27	209.48	141.22	55.148	82.941	193.01	81.526	33.537	53.262
Avg. MSE	433.47	299.47	165.01	422.07	427.9	258.37	118.66	191.69	407.08	174.35	87.88	122.13	381.91	110.94	58.691	87.64

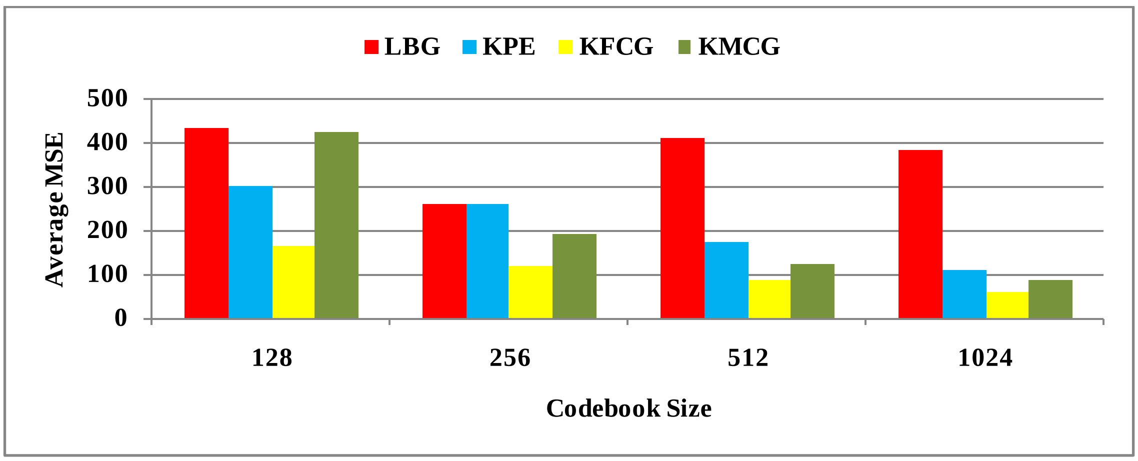


Figure 8. Average MSE obtained using all four clustering algorithms with Codebook sizes 128, 256, 512 and 1024 for car category.

Table 2. Shows the MSE for the images in Training set for facial category using all four algorithms for different Codebook sizes 128, 256, 512 and 1024. Figure 9. Shows the Average MSE obtained using all four clustering algorithms with Codebook sizes 128, 256, 512 and 1024 for facial category. It is observed that for the facial category for all the codebook sizes 128 to 1024 the MSE obtained from the KFCG is less as compared to LBG, KPE and KMCG.

Table 2. MSE for the images in Training set for facial category using all four algorithms for different Codebook sizes 128, 256, 512 and 1024.

Image No.	Codebook size 128				Codebook size 256				Codebook size 512				Codebook size 1024			
	LBG	KPE	KFCG	KMCG	LBG	KPE	KFCG	KMCG	LBG	KPE	KFCG	KMCG	LBG	KPE	KFCG	KMCG
1	167.21	170.73	50.795	97.931	158.89	134.86	36.035	59.139	155.14	85.536	26.717	32.634	140.39	57.622	19.154	23.306
2	161.52	163.65	66.087	104.01	156.52	123.58	51.674	63.804	154.16	87.409	38.369	47.842	139.13	61.138	27.415	36.515
3	297.35	316.2	63.123	102.53	288.12	234.37	49.178	61.323	283.26	118.34	36.933	46.644	270.98	60.288	25.792	32.766

4	96.671	93.135	50.94	73.579	93.228	77.106	41.243	56.242	89.841	55.829	32.982	38.732	79.641	37.787	23.858	29.729
5	165.56	157.81	55.287	115.8	161.9	127.25	43.457	59.31	157.7	74.925	35.77	41.328	138.93	48.573	23.367	28.037
6	119.21	163.7	47.815	67.964	134.96	124.64	34.538	38.272	114.13	37.207	25.25	33.041	147.47	30.078	18.872	23.782
7	273.16	319.17	49.009	56.573	266.87	162.29	36.936	44.268	278.52	56.505	29.376	35.872	263.81	38.401	22.373	27.478
8	194.57	195.14	139.59	173.27	191.22	179.34	107.21	122.62	185.48	156.05	80.953	98.023	174.02	124.04	60.669	78.635
9	437.38	434.78	262.64	336.97	431.79	392.18	192.04	259.85	424.11	335.92	150.6	201.1	404.7	262.43	118.74	159.77
10	117.93	120.21	78.74	82.913	112.85	100.59	58.828	62.604	108.57	76.972	42.847	50.011	101.52	63.059	32.064	40.558
Avg. MSE	203.06	213.45	86.40	121.15	199.63	165.62	65.11	87.7432	195.09	108.47	49.98	62.5227	186.06	78.34	37.23	48.057

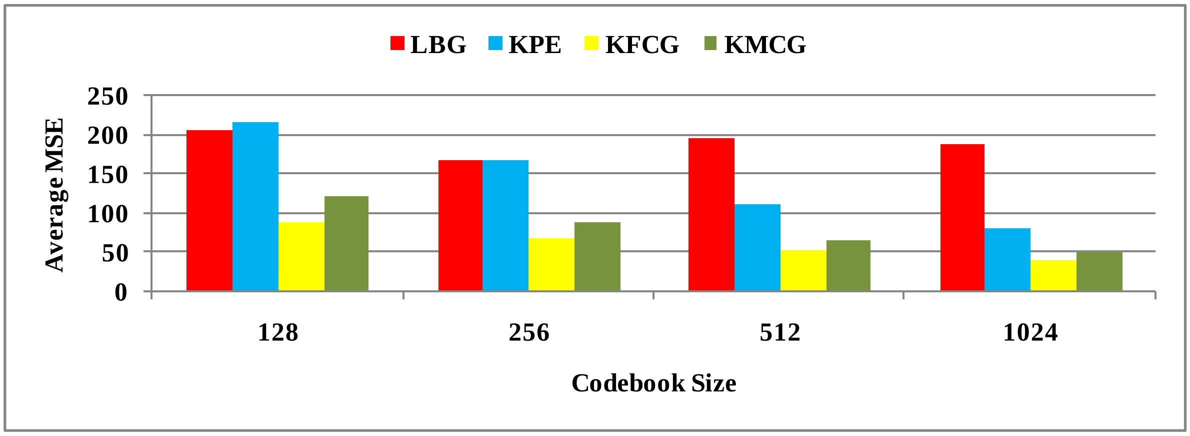


Figure 9. Average MSE obtained using all four clustering algorithms with Codebook sizes 128, 256, 512 and 1024 for facial category.

Table 3. and Table 4. Shows the MSE of all the five images in the Cars and Facial Test Set using all four algorithms for different Codebook sizes 128, 256, 512 and 1024 respectively.

Table 3. MSE of images in the Cars Test Set using all four algorithms for different Codebook sizes 128, 256, 512 and 1024.

Image No.	Codebook size 128				Codebook size 256				Codebook size 512				Codebook size 1024			
	LBG	KPE	KFCG	KMCG	LBG	KPE	KFCG	KMCG	LBG	KPE	KFCG	KMCG	LBG	KPE	KFCG	KMCG
1	697.5	442.67	228.95	658.63	593.66	318.17	168.48	464.1	549.75	217.03	130.06	224.2	477.04	139.18	99.2	139.67
2	543.49	405.13	190.4	752.3	522.43	320.67	127.62	259.87	492.35	201.47	90.362	131.27	419.03	107.72	50	80.679
3	945.79	596.53	273.75	941.12	869.47	415.23	200.58	308.58	818.31	280.33	153.75	198.65	707.24	165.49	98.26	138.15
4	346.67	271.51	157.97	296.2	326.41	208.46	107.47	159.66	304.65	143.44	81.29	103.04	273.52	79.548	47.17	67.2
5	403.06	386.52	210.16	663.62	387.69	323.07	146.33	228.58	361.37	226.95	103	141.65	326.96	120.31	60.24	83.554
Avg. MSE	587.30	420.47	212.25	662.37	539.93	317.12	150.10	284.16	505.29	213.84	111.69	159.76	440.76	122.45	70.97	101.850

Figure 10. Shows the Results of all five test images belonging to car category using Global codebook obtained from LBG, KPE, KFCG and KMCG for the codebook sizes 1024x12.

.

Original LBG KPE KFCG KMCG

Figure 10. Results of all five test images belonging to car category using Global codebook obtained from LBG, KPE, KFCG and KMCG for the codebook sizes 1024x12.

Table 4. MSE of images in the Facial Test Set using all four algorithms for different Codebook sizes 128, 256, 512 and 1024.

Image No.	Codebook size 128				Codebook size 256				Codebook size 512				Codebook size 1024			
	LBG	KPE	KFCG	KMCG	LBG	KPE	KFCG	KMCG	LBG	KPE	KFCG	KMCG	LBG	KPE	KFCG	KMCG
1	132.81	131.06	71.003	105.88	122.73	115.84	57.23	75.351	116.03	92.877	46.252	61.797	103.65	61.923	35.509	48.043
2	91.478	101.79	38.192	74.767	107.39	75.435	30.774	49.789	82.357	49.963	23.792	36.941	97.071	30.483	17.976	25.776
3	131.15	130.18	59.223	89.297	116.65	110.38	46.007	55.379	112.62	85.72	36.484	45.826	99.312	53.543	27.34	36.546
4	134.41	131.51	66.138	97.541	127.19	117.89	46.991	62.992	118.18	97.658	34.106	58.181	107.75	64.251	22.652	45.98
5	129.34	127.66	66.01	87.702	113.8	101.42	52.845	65.846	111.66	76.219	40.255	50.761	94.97	51.234	30.149	39.726
Average MSE	123.84	124.44	60.11	91.037	117.55	104.19	46.77	61.871	108.17	80.49	36.18	50.701	100.55	52.29	26.73	39.214

Figure 11. Shows the Results of all five test images belonging to facial category using Global codebook obtained from LBG, KPE KFCG and KMCG for the codebook sizes 1024x12.

Original LBG KPE KFCG KMCG

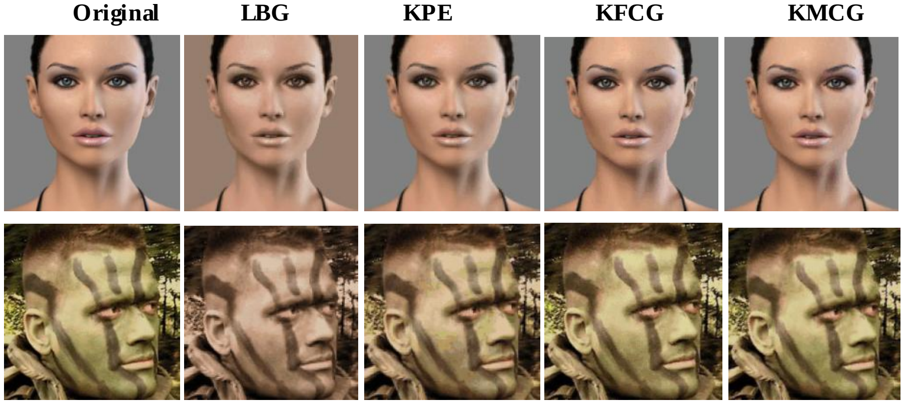


Figure 11. Results of all five test images belonging to facial category using Global codebook obtained from LBG, KPE, KFCG and KMCG for the codebook sizes 1024x12.

5. CONCLUSION

This paper presents the Global codebook generation algorithms using LBG, KPE, KMCG and KFCG clustering algorithm. From results it is clearly observe that the decoded images from the KFCG algorithm with codebook sizes 128 to 1024 are of very high quality, almost close to the original image having very low MSE. With a larger training set and even larger codebook, the MSE for the training set as well as test set reduce even further and allow real-time transfer of highly compressed data only using one global codebook. But the limitation of a very large codebook will be it will lead to higher computing time which may degrade the real-time performance. It is also noticed that the degree of similarity between images in the test set and training set has a huge impact on the MSE. The Global Codebook Technique will be extremely efficient under the following stipulations:

1. Use the KFCG algorithm with sufficiently large codebook size.
2. Use similar images with color variations to ensure low loss during real-time compression.
3. Ensure the images being compressed in real-time are similar to as many images as possible in the training set.

6. REFERENCES

[1] Jeng-Shyang Pan, Zhe-Ming Lu, and Sheng-He Sun.: ‘An Efficient Encoding Algorithm for Vector Quantization Based on Subvector Technique’, IEEE Transactions on image processing, vol 12 No. 3 March 2003.

[2] R. M. Gray.: ‘Vector quantization’, IEEE ASSP Mag., pp. 4-29, Apr.1984.

[3] Y. Linde, A. Buzo, and R. M. Gray.: ‘An algorithm for vector quantizer design,” IEEE Trans. Commun.’, vol. COM-28, no. 1, pp. 84-95, 1980.

[4] Gersho, R.M. Gray.: ‘Vector Quantization and Signal Compressio’, Kluwer Academic Publishers, Boston, MA, 1991.

[5] Chin-Chen Chang, Wen-Chuan Wu, “Fast Planar-Oriented Ripple Search Algorithm for Hyperspace VQ Codebook”, IEEE Transaction on image processing, vol 16, no. 6, June 2007.

[6] Momotaz Begum, Nurun Nahar, Kaneez Fatimah, M. K. Hasan, and M. A. Rahaman: ‘An Efficient Algorithm for Codebook Design in Transform Vector Quantization’, WSCG’2003, February 3-7, 2003.

[7] Robert Li and Jung Kim: ‘Image Compression Using Fast Transformed Vector Quantization’, 29th Applied Imagery Pattern Recognition Workshop, 2000, pp. 141 – 145, Apr. 2000.

[8] Zhibin Pan; Kotani, K.; Ohmi, T., ‘Enhanced fast encoding method for vector quantization by finding an optimally-ordered Walsh transform kernel’, ICIP 2005, IEEE International Conference, Volume 1, Issue, 11-14, Page(s): I - 573-6, Sept. 2005.

[9] Jim Z.C. Lai, Yi-Ching Liaw, and Julie Liu, “A fast VQ codebook generation algorithm using codeword displacement”, Pattern Recogn. vol. 41, no. 1, pp 315–319, 2008.

[10] Y.C. Liaw, J.Z.C. Lai, W. Lo, Image restoration of compressed image using classified vector quantization, Pattern Recogn. vol. 35, No. 2, pp 181–192, 2002.

[11] N.M. Nasrabadi, Y. Feng, Image compression using address vector quantization, IEEE Trans. Commun. vol. 38 No. 12, pp. 2166–2173, 1990.

[12] J. Foster, R.M. Gray, M.O. Dunham, Finite state vector quantization for waveform coding, IEEE Trans. Inf. Theory vol. 31, No. 3, pp. 348–359, 1985.

[13] T. Kim, Side match and overlap match vector quantizers for images, IEEE Trans. Image Process. vol. 1, No. 2, pp. 170–185, 1992.

[14] J. Z. C. Lai, Y.C. Liaw, W. Lo, Artifact reduction of JPEG coded images using mean-removed classified vector quantization, Signal Process. vol. 82, No. 10, pp. 1375–1388, 2002.

[15] K. N. Ngan, H.C. Koh, Predictive classified vector quantization, IEEE Trans. Image Process. vol. 1, No. 3, pp. 269–280, 1992

[16] S. C. Lo, H. P. Chan, J. S. Lin, H. Li, M. T. Freedman, and S. K. Mun, "Artificial convolution neural network for medical image pattern recognition," Neural Networks, vol.8, no. 7/8, pp. 1201–1214, 1995.

[17] Y. Linde, A. Buzo, and R. M. Gray, "An algorithm for vector quantizer design," IEEE Trans.Commun., vol. COM- 28, no. 1, pp.: 84-95, 1980

[18] H.B.Kekre, Tanuja K. Sarode, "New Fast Improved Clustering Algorithm for Codebook Generation for Vector Quantization", International Conference on Engineering Technologies and Applications in Engineering, Technology and Sciences, Computer Science Department, Saurashtra University, Rajkot, Gujarat. (India), Amoghsiddhi Education Society, Sangli, Maharashtra (India) , 13th – 14th January 2008.

[19] H. B. Kekre, Tanuja K. Sarode, "New Fast Improved Codebook Generation Algorithm for Color Images using Vector Quantization," International Journal of Engineering and Technology, vol.1, No.1, pp.: 67-77, September 2008.

[20] H. B. Kekre, Tanuja K. Sarode, "Fast Codebook Generation Algorithm for Color Images using Vector Quantization," International Journal of Computer Science and Information Technology, Vol. 1, No. 1, pp.: 7-12, Jan 2009.

[21] H. B. Kekre, Tanuja 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, pp.: 62- 67, 16-18 July 2008. Avaliable at IEEE Xplore.

[22] H. B. Kekre, Tanuja K. Sarode, "Fast Codebook Generation Algorithm for Color Images using Vector Quantization," International Journal of Computer Science and Information Technology, Vol. 1, No. 1, pp.: 7-12, Jan 2009.

[23] H. B. Kekre, Tanuja K. Sarode, "Fast Codevector Search Algorithm for 3-D Vector Quantized Codebook", WASET International Journal of cal Computer Information Science and Engineering (IJCISE), Volume 2, No. 4, pp.: 235-239, Fall 2008. Available: http://www.waset.org/ijcise.

[24] H. B. Kekre, Tanuja K. Sarode, "Fast Codebook Search Algorithm for Vector Quantization using Sorting Technique", ACM International Conference on Advances in Computing, Communication and Control (ICAC3-2009), pp: 317-325, 23-24 Jan 2009, Fr. Conceicao Rodrigous College of Engg., Mumbai. Available on ACM portal.

[25] Jim Z.C. Lai, Yi-Ching Liaw, and Julie Liu, "A fast VQ codebook generation algorithm using codeword displacement", Pattern Recogn. vol. 41, no. 1, pp.: 315–319,2008.

[26] C.H. Hsieh, J.C. Tsai, Lossless compression of VQ index with search order coding, IEEE Trans. Image Process. vol. 5, No. 11, pp.: 1579–1582, 1996.

[27] Chin-Chen Chang, Wen-Chuan Wu, "Fast Planar-Oriented Ripple Search Algorithm for Hyperspace VQ Codebook", IEEE Transaction on image processing, vol 16, no. 6, pp.: 1538-1547, June 2007.

[28] Garcia and G. Tziritas, "Face detection using quantized skin color regions merging and wavelet packet analysis,"= IEEE Trans. Multimedia, vol. 1, no. 3, pp.: 264–277, Sep. 1999.

[29] H. Y. M. Liao, D. Y. Chen, C. W. Su, and H. R. Tyan, "Real-time event detection and its applications to surveillance systems," in Proc. IEEE Int. Symp. Circuits and Systems, Kos, Greece, pp.: 509–512, May 2006.

[30] J. Zheng and M. Hu, "An anomaly intrusion detection system based on vector quantization," IEICE Trans. Inf. Syst., vol. E89-D, no. 1, pp.: 201–210, Jan. 2006.

[31] H. B. Kekre, Tanuja K. Sarode, Bhakti 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, 23-24 Jan 2009, Fr. Conceicao Rodrigous College of Engg., Mumbai. Available on ACM portal.

[32] H. B. Kekre, Tanuja K. Sarode, Bhakti Raul, "Color Image Segmentation using Kekre's Algorithm for Vector Quantization", International Journal of Computer Science(IJCS), Vol. 3, No. 4, pp.: 287-292,Fall2008. Available:http://www.waset.org/ijcs.

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

[34] H. B. Kekre, Tanuja K. Sarode, Bhakti Raul, "Color Image Segmentation Using Vector Quantization Techniques",Advances in Engineering Science Sect. C (3), pp.: 35-42,July-September 2008.

[35] H. B. Kekre, Tanuja K. Sarode, "Speech Data Compression using Vector Quantization", WASET International Journal of Computer and Information Science and Engineering (IJCISE), vol. 2, No. 4, pp.: 251-254, Fall 2008.

[36] H. B. Kekre, Ms. Tanuja K. Sarode, Sudeep 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), Volume 9, Issue 5, pp.: 1-8, September 2009.

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

7. AUTHOR BIOGRAPHIES

Dr. H. B. Kekre has received B.E. (Hons.) in Telecomm. Engineering. from Jabalpur University in 1958, M.Tech (Industrial Electronics) from IIT Bombay in 1960, M.S.Engg. (Electrical Engg.) from University of Ottawa in 1965 and Ph.D. (System Identification) from IIT Bombay in 1970 He has worked as Faculty of Electrical Engg. and then HOD Computer Science and Engg. at IIT Bombay. For 13 years he was working as a professor and head in the Department of Computer Engg. at Thadomal Shahani Engineering. College, Mumbai. Now he is Senior Professor at MPSTME, SVKM's NMIMS. He has guided 17 Ph.Ds, more than 100 M.E./M.Tech and several B.E./ B.Tech projects. His areas of interest are Digital Signal processing, Image Processing and Computer Networking. He has more than 270 papers in National / International Conferences and Journals to his credit. He was Senior Member of IEEE. Presently He is Fellow of IETE and Life Member of ISTE Recently seven students working under his guidance have received best paper awards. Currently 10 research scholars are pursuing Ph.D. program under his guidance.

Dr. Tanuja K. Sarode has Received Bsc.(Mathematics) from Mumbai University in 1996, Bsc.Tech.(Computer Technology) from Mumbai University in 1999, M.E. (Computer Engineering) degree from Mumbai University in 2004, Ph.D. from Mukesh Patel School of Technology, Management and Engineering, SVKM's NMIMS University, Vile-Parle (W), Mumbai, INDIA. She has more than 11 years of experience in teaching. Currently working as Assistant Professor in Dept. of Computer Engineering at Thadomal Shahani Engineering College, Mumbai. She is life member of IETE, member of International Association of Engineers (IAENG) and International Association of Computer Science and Information Technology (IACSIT), Singapore. Her areas of interest are Image Processing, Signal Processing and Computer Graphics. She has more than 100 papers in National /International Conferences/journal to her credit.

Ms. Kavita C.Raut has received B.E. (Computer Engg.) degree from Mumbai University in 2005, currently Pursuing M.E. from Thadomal Shahani Engineering College, Bandra (W), Mumbai. She has more than 5 years of experience in teaching. She is currently working as a lecturer in department of Computer Engineering at Thadomal Shahani Engineering College, Mumbai. She has 4 papers in National /International journal to her credit.

Rohan Tahiliani is currently pursuing M.S. in Information Security from Georgia Institute of Technology. He aims to complete his M.S. and Ph.D. degree in Network Security. His area of interest is System and Network Security and Information Hiding.