A Novel Codebook Design Algorithm for Image Vector Quantization

V.Manohar*, D.Harikrishna^1, Chiranjeevi Karri^2, M.Ranjeeth^3
Assistant Professor, Vaagdevi Engineering College,Warangal,Telangana, India
^{1}Associate Professor, B.V.Raju Institute of Technology,Narsapur,Medak, Telangana, India,
^{2}Post Doctoral Researcher,Faculty of Engineering,University of Porto, Portugal
^{3}Assistant Professor,Vaagdevi College of Engineering,Warangal,Telangana,India
**manoharvu@gmail.com, ^{1}harikrishna.dodde@bvrit.ac.in, ^{2}chiru404@gmail.com, ^{3}ranjithmamidi2001@gmail.com*

Abstract
Codebook designation is a major issue in Vector quantization (VQ). The Linde-Buzo-Gray (LBG) algorithm is the traditional method for VQ codebook generation. The initial codebook selection has little effect on the LBG algorithm's efficiency. However, the LBG algorithm is frequently trapped in a local minimum of distortion and always generates a local optimal codebook, resulting in random codebook initialization. An efficient codebook initialization algorithm based on Kernel density estimation is proposed in this paper. The proposed technique is correlated to the LBG as well as five evolutionary optimization algorithms: Particle Swarm Optimization (PSO-LBG), Quantum PSO (QPSO-LBG), Honey Bee Mating Optimization (HBMO-LBG), Bat algorithm (BA-LBG) and Firefly algorithm (FA-LBG). According to the experimental outcomes, the proposed technique performs better than the other six algorithms. Furthermore, The quality of the restored images also gets better.

Keywords: *Vector Quantization, Codebook genaration, VQ Codebook Design, Generalized Lloyd Algorithm, Kernal Density Estimation.*

1. Introduction

Digital images have become increasingly popular as electronic devices such as digital cameras and mobile devices have evolved. Because of their large size, digital images require a lot of storage space and fast transmission rates. Because mobile devices have limited memory and use wireless networks that typically have bandwidth limitations, these constraints limit the use of digital images. Image compression techniques have developed and are used to convert digital images in compressed form, which decreases the storage space required and saves bandwidth, in order to make digital image transmission compatible with these limitations. This will also speed up transmission.

In the last two decades, many image compression techniques have been proposed. These methods are divided into *lossless* and *lossy* compression. The decompressed image in lossless image compression techniques is identical to the original image; however, lossy image compression reconstructs an approximation of the image from the compressed image without visual distortion. Lossy compression techniques clearly have a higher compression rate than lossless techniques. Frequently used lossy methods are those based on discrete wavelet transformation (DWT) [1, 2], vector quantization (VQ) [3, 4], and discrete cosine transformation (DCT). The techniques described above can reduce image redundancy and, as a result, image size. VQ, in particular, is an efficient technique that is more suitable for low power computation systems such as mobile devices due to its simple structure, fast decoding capability, and more compression rate. VQ can also be used in a variety of applications, including pattern recognition [5,6], speech recognition [7], data hiding [8,9], and image authentication [10]. There are three major steps in the VQ process: i) Codebook generation ii) Encoding iii) Decoding. A codebook requires for the encoding and decoding processes. The Generalized Lloyd Algorithm (GLA) [11] is commonly utilized method for codebook generation. GLA, also called as Linde-Buzo-Gary (LBG), begins with an initial solution that is improved sequentially using two optimality criteria until a local minimum is reached. The initial codebook selection has little effect on the LBG's performance. The first codebook in conventional LBG is chosen at random from the training data set. It always converges to the nearest local minimum due to non-optimal codebook initialization.

Over the last two decades, evolutionary optimization algorithms have used to design codebooks in order to improve the results of the LBG algorithm. Chen, Yang, and Gou [12] improved the LBG codebook design algorithm based on the PSO algorithm.The global best particle was initialised using the LBG algorithm's output, which sped up the PSO's convergence. A QPSO was also suggested by Wang et al. to resolve the 0-1 knapsack issue [13]. A new vector quantization approach based on honey bee mating optimization was further extended by Horng and Jiang (HBMO)[14]. In comparison to the LBG, PSO- LBG, and QPSO- LBG algorithms, HBMO produced a low-distortion, high-quality reconstructed image. Furthermore, other evolutionary optimization algorithms, such as FA [15] and the BA [16], are proposed and improved the codebook quality of the vector quantization algorithm. The codebooks are optimised by an improved optimization algorithm as a novel intention [18].for the production of codebooks in VQ, a new DPIO algorithm. The DPIO-LBG process is the name given to the model, which uses LBG to initialise the DPIO algorithm to develop the VQ approach [19].

Though evolutionary optimization algorithms reduce the possibility of convergence to the local minimum due to their parallel search structure, these algorithms have a high computational complexity. As a result, these algorithms require a huge amount of computation time and memory space (especially when the population size increases).By utilising the statistical features of training data, a novel technique for enhancing codebook design quality is presented in this paper. The initial cluster centres in the proposed approach are chosen based on the amount of data present in each Euclidean space region. The Parzen windowing method is used to estimate the distribution function of the magnitude of the training vectors for this purpose. The proposed algorithm is faster than compared evolutionary optimization algorithms because it uses statistical properties rather than probabilistic search. The remaining parts are ordered as follows:

1. Basic ideas are presented in Section
2. The proposed approach is demonstrated in Section
3. Section presents experimental findings
4. Section ends the analysis.

2. Basic Concepts

2.1. Vector quantization

There are three major steps in the VQ process: codebook generation, encoding, and decoding. The image is first divided into non-overlapping $p \times q$ blocks during the encoding procedure. Then for each block, a k dimensional vector is created with $k = p \times q$. These vectors are referred to as training vectors, and the set of training vectors is referred to as the training set. Encoder detects the nearest codeword for every training vector based on

Euclidean distance, assuming the codebook was created previously. The following equation can be used to calculate the Euclidean distance across the training vector 'x' and the codeword 'c_i'

$$Euclidean\ distance = \|x - c_i\| = \sum(x_j - c_{ij}) \qquad (1)$$

The jth components of the training vector x and the codeword c_i are represented by x_j and c_{ij} in the above equation.

Following the detection of the nearest codeword, The index associated with that codeword should be saved in the index table. This process is performed again each training vector, and the resulting index table is sent to the decoding section. The codeword associated with each training vector is extracted from the codebook using the index table during the decoding process, and the image is reconstructed.

2.2. Generalized Lloyd Algorithm

The GLA algorithm known as the Linde-Buzo-Gary (LBG) algorithm and commonly generates codebook for VQ. The following are the steps of this algorithm:

1. First, N vectors should be randomly chosen as primary codeword's among the data set $X = x_i$, where i = 1, 2,3,….., M and codewords are shown as: $c_1, c_2, c_3 \ldots\ldots c_N$.

2. Using the square of the Euclidean distance, identify the closest codeword for every vector in the data set (X). The associated cluster is then expanded with the vector using the formula below:

$$x_i = \|x_i - c_j < x_i - c_p\| \qquad (i, j, p = 1,2,3 \ldots\ldots\ldots N) \qquad (2)$$

Where, || . || = Euclidean distance, $x_i = i_{th}$ cluster, $c_j = j_{th}$ cluster, $c_p = p_{th}$ cluster.

3. The following equation will be used to determine the new codewords:

$$c_i = \frac{1}{s_i}\sum x_j \qquad \text{where, } j\in 1,2\ldots\ldots N, \quad i\in 1,2\ldots\ldots\ldots M \qquad (3)$$

Where s_i = No. of Vectors that relates to group of x_j

4. The iteration should be terminated: if the difference between codeword's in two consecutive iterations is less than a certain threshold, otherwise if certain no. of iterations has reached.

The LBG algorithm's dependence on the initial codebook is one of its most significant drawbacks. This means that if the first and final codebook is ineffective, poor quality. LBG always converges to the nearest local minimum due to random codebook intialization.

2.3. Kernel density estimation (KDE)

KDE is a nonparametric method for estimating a random variable's probability density function [17]. Nonparametric estimators have no consistent structure and must rely on all of the data points to produce an estimate, in contrast to parametric estimators, that have a constant functional structure and only need to store the elements of that function.

Let X is the random variable with a unknown density function that is $f(x)$. Parzen-windowing estimates the pdf of random variable X using its samples ($X_1, X_2,\ldots, X_N$). The following equation is used to calculate the kernel density estimator for X:

$$f_{(x)} = \frac{1}{N}\sum_{i=1}^{N} k_h(\mathrm{X} - X_i) = \frac{1}{N_h}\sum_{i=1}^{N} k\left(\frac{\mathrm{X} - X_i}{h}\right) \qquad (4)$$

Where: k(.) represents the kernel function and
h indicates a smoothing parameter (bandwidth).

The scaled kernel is a kernel with subscript h. commonly used kernel functions include uniform, triangular, normal, and others. The normal (Gaussian) kernel is frequently used due to its convenient mathematical properties:

$$\mathrm{k(x)} - \mathrm{G(x)} = \frac{1}{\sqrt{2\Pi}} e^{\frac{-x^2}{2}} \tag{5}$$

The degree of smoothing is controlled by the bandwidth h, which has a significant impact on the resulting estimate. Figure 1. shows estimated probability density function of Euclidean lengths for training vectors of set I. The resulting curve is irregular and contains an excessive amount of erroneous data distortions when h is narrow. However, when h is too high, significant characteristics are lost.

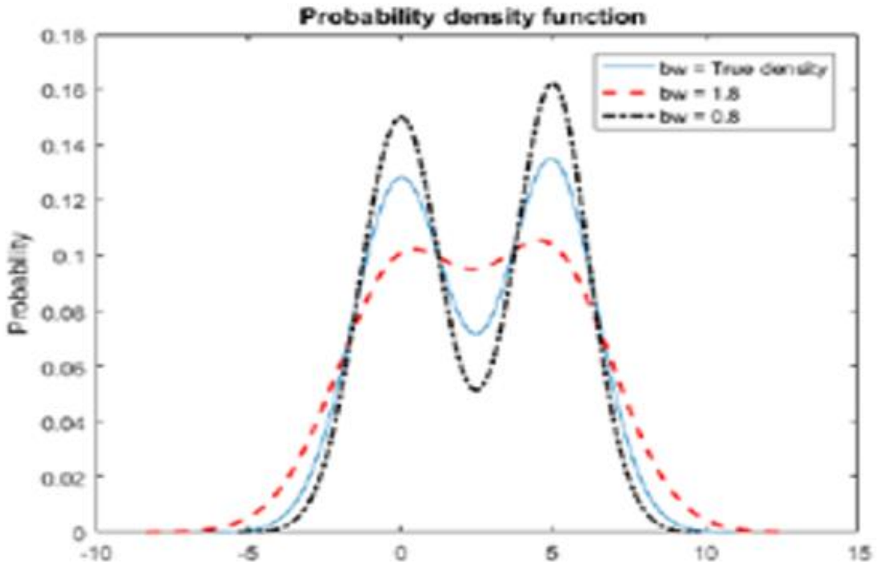

Figure 1. Kernel density estimate (KDE) with different bandwidths

3. Proposed Method

A new technique for codebook initialization is proposed in this section. In this Technique, K dimensional space is segmented into non-overlapping areas, and the initial cluster centres are choosen depends on data density in each areas.

We use the Euclidean norm measure to divide the space into non-overlapping areas. In *k-dimensional space*, the Euclidean norm of a vector is calculated as follows:

$$\|x\| = \sqrt{\sum_{j=0}^{k} x_j^2} \tag{6}$$

An Euclidean vector is the position of a point in Euclidean k-dimensional space. Consider Figure 2, which depicts a 2-dimensional data set of training vectors. This diagram displays the length and width of the petals from the Fisher's Iris data collection, this data set is called as set I.

According to their magnitudes, space areas are divided into 8 parts in Figure 2. Data density is not equally distributed between segments, as can be shown. The best choice is therefore not to choose the initial cluster centres at random. Initial cluster centres should be selected depending on data density in each location to provide a more efficient selection.

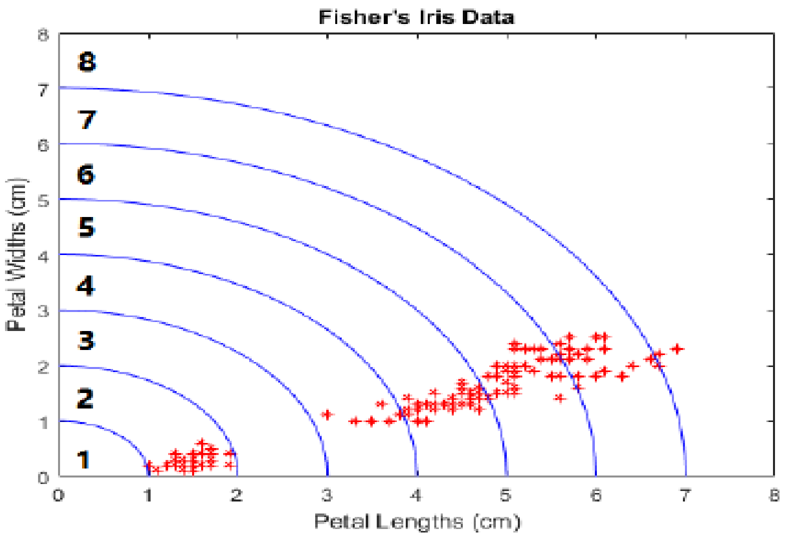

Figure 2. For the training vectors of set I, an analysis of data density in various locations of two-dimensional space.

The Parzen windowing approach with a kernel function should be used to estimate the distribution function of the magnitude of the training vectors in order to achieve data distribution in various locations. The calculated probability density function for Euclidean lengths for set I training vectors is that by integrating the probability density function of Euclidean lengths, f(r) on various regions, the amount of data present in each area can be estimated.Each region's initial cluster centre count need to be proportional to its data density.

Table 1.Data density in various areas for set I training vectors

Region	Euclidean norm(cm)	Data density
1	$R < 1$	0 %
2	$1 < R < 2$	32 %
3	$2 < R < 3$	0 %
4	$3 < R < 4$	5 %
5	$4 < R < 5$	22 %
6	$5 < R < 6$	23 %
7	$6 < R < 7$	14 %
8	$7 < R$	4 %

Table 1 displays the data density for each region for training vectors in set I. As a result, The following are the major image compression steps for our proposed method:

- Divide the image into small 4 x 4 blocks.
- Create the size N training set.
- The Euclidean length of each training vector in the training set should be determined.
- Using the Parzen windowing method, estimate the distribution function of the magnitude of the training vectors.

- Segment space areas depend on Euclidean length and integrate pdf of Euclidean norms on various areas.
- Select initial cluster centres in each region based on data density.
- Utilize the 'LBG method' to build final Codebook using the initial Codebook created in the previous stage as input.
- Use final codebook to obtain the index table.

4. Experimental results

A computational experiment has accomplished to assess the efficacy of the suggested technique. The simulations are run on to a computer with an Intel Core i5-4200M processor running at 2.50 GHz and 6GB of RAM. Furthermore, all simulations are compiled with the MATLAB software version 7.9.0. (R2009b). We choose three benchmark images with pixel amplitude resolutions of *8bits* to compare the proposed algorithm to other algorithms: "BABOON", "LENA" and "PEPPER". These are gray scale images with 512× 512 pixels in size, as shown in Figure 3. LBG, PSO-LBG, QPSO-LBG, HBMO-LBG, FA-LBG, BA-LBG, and the proposed algorithm are used to compress these images.

MSE - Mean Square Error, PSNR - Peak Signal to Noise Ratio, and bit-rate are the metrics used to compare the proposed method to other methods, as shown in Eqs. (7), (8), and (9). The PSNR is the peak error. The PSNR metric represents the encoded image's quality. These metrics are computed for all images with codebook sizes ranging from 16, 32, 64, 128, 256, and 512.

$$MSE = \frac{1}{M \times N} \sum_{X}^{M} \sum_{Y}^{N} \{ f(x,y) - \bar{f}(x,y) \}^2 \tag{7}$$

Where, M × N is the size of the input image .

$$PSNR = 10 \times log_{10} \left(\frac{255^2}{MSE} \right) \; dB \tag{8}$$

$$bit - rate = \; log_2 \, C_K \tag{9}$$

Where, $C_K =$ Size of the Codebook.

(i) (ii) (iii)

Figure 3. Three test images: (i) Lena (ii) Baboon (iii) Pepper

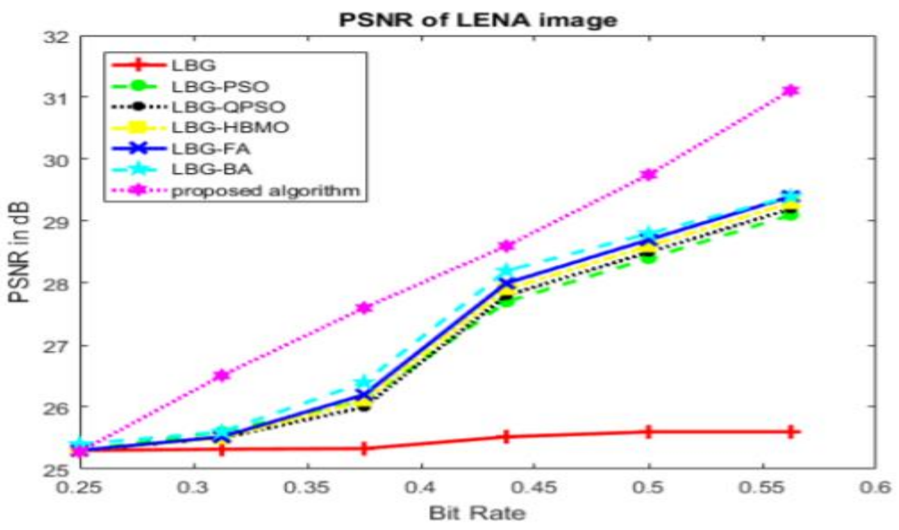

Figure 4.The average PSNR of an Lena image using various algorithms.

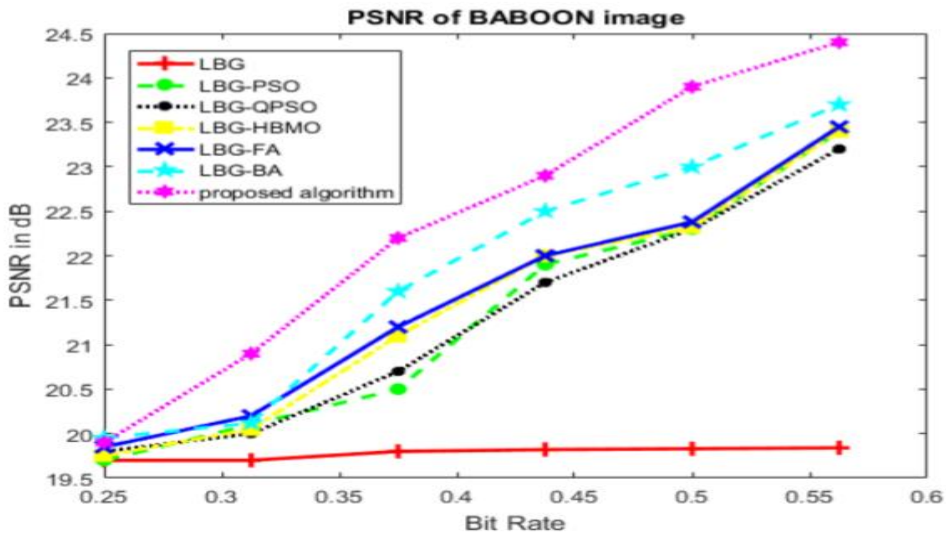

Figure 5.The average PSNR of Baboon image using various algorithms.

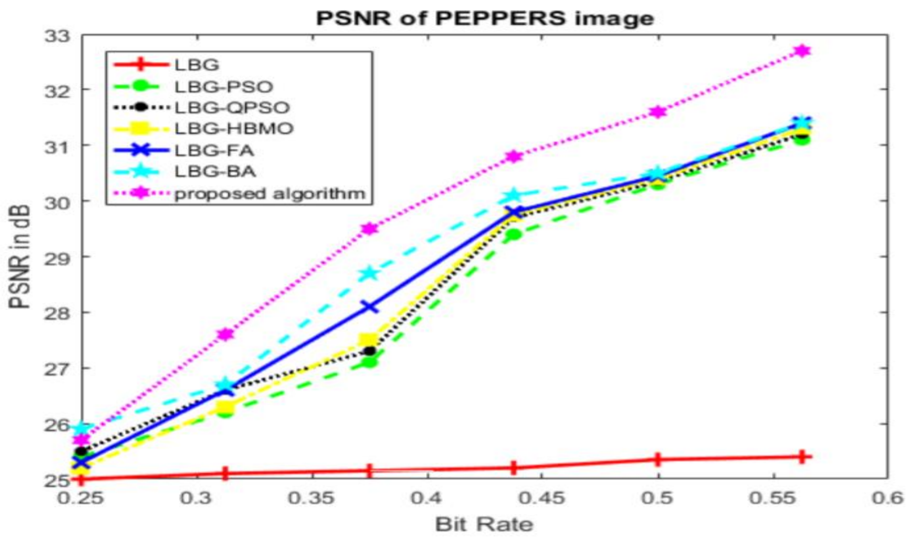

Figure 6.The average PSNR of Pepper image using various algorithms.

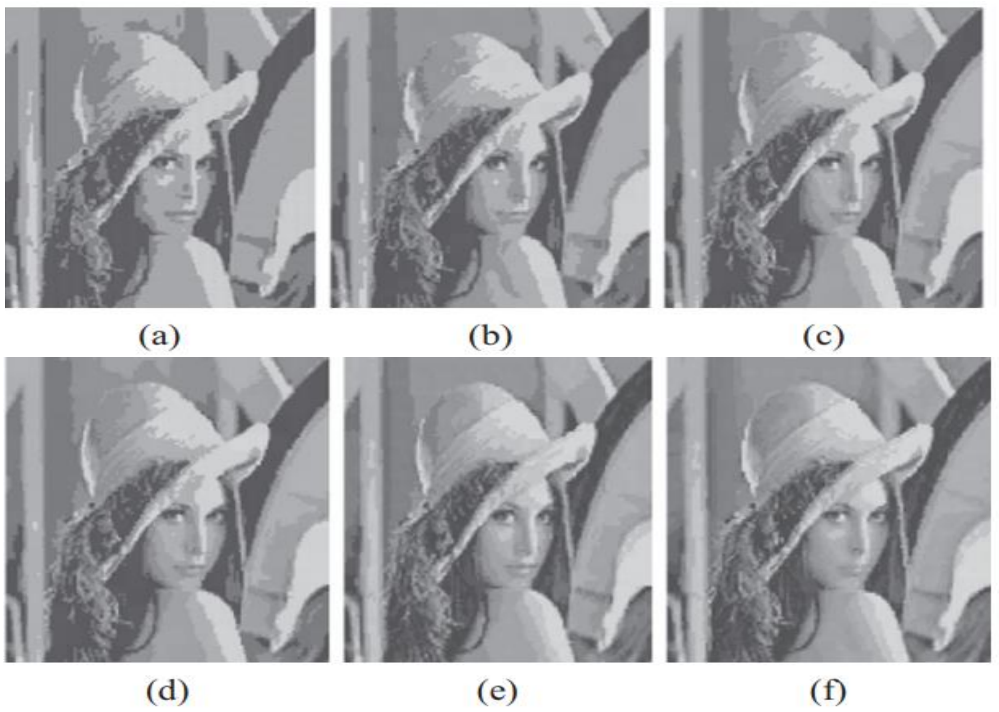

Figure 7. Decompressed Lena image using various algorithms (Codebook size = 256): (a) LBG (b) PSO- LBG, (c) HBMO- LBG (d) FA- LBG (e) BA-LBG (f) Proposed algorithm

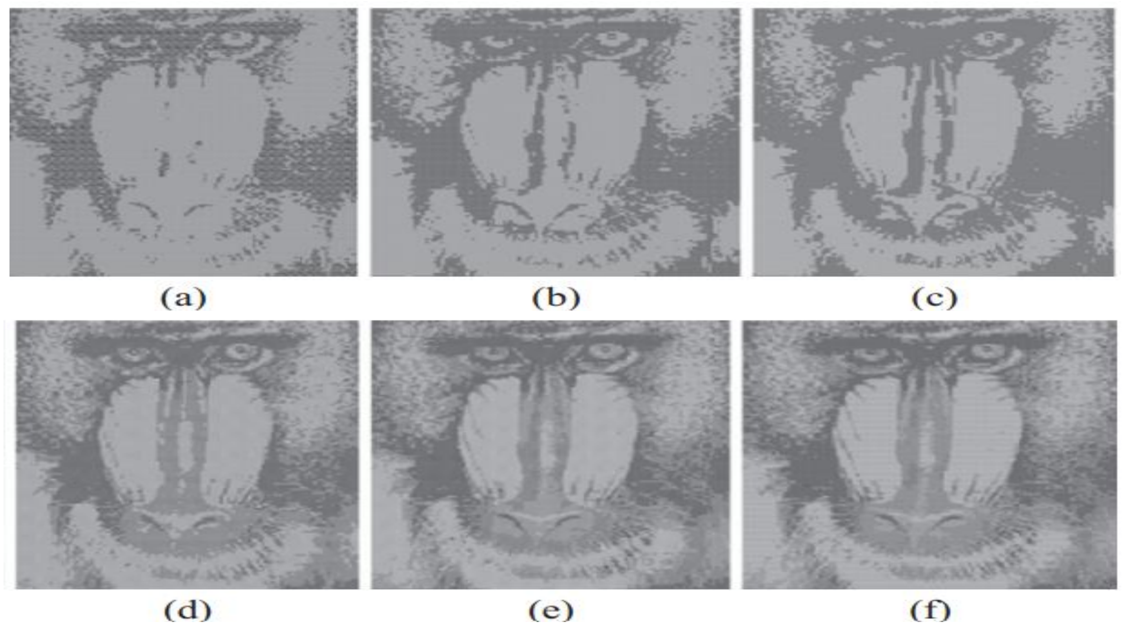

Figure 8. Decompressed Baboon image for different algorithms (codebook size = 256): (a) LBG (b) PSO- LBG, (c) HBMO- LBG (d) FA- LBG (e) BA- LBG (f) Proposed algorithm

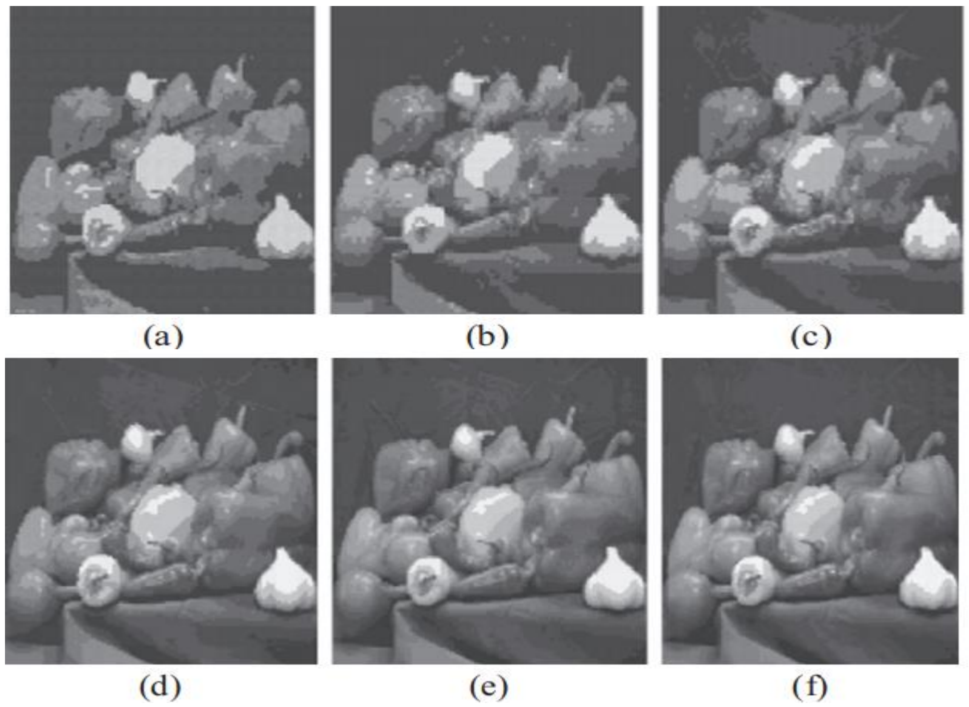

Figure 9. Decompressed Pepper image for different algorithms (codebook size = 256): (a) LBG (b) PSO- LBG, (c) HBMO- LBG (d) FA- LBG (e) BA- LBG (f) proposed algorithm

Table 2. The average computation time (sec) for various vector quantization algorithms (bit rate = 0.2500)

Codebook size: 16							
Image	LBG	PSO-LBG	QPSO-LBG	HBMO - LBG	FF-LBG	BA-LBG	Proposed
Lena	3.40	251.88	266.20	528.99	507.18	253.03	4.79
Pepper	4.57	249.81	252.43	566.65	534.30	325.78	5.98
Baboon	4.53	320.69	332.84	951.32	943.36	334.66	5.95
Avg.	4.16	274.12	283.82	682.32	661.61	304.49	5.57

Table 3. The average computation time (sec) for various vector quantization algorithms (bit rate = 0.3125)

Codebook size: 32							
Image	LBG	PSO-LBG	QPSO-LBG	HBMO - LBG	FF-LBG	BA-LBG	Proposed
Lena	5.26	305.53	317.31	760.34	711.81	344.74	9.62
Pepper	6.24	337.61	271.90	572.87	593.78	348.99	6.60
Baboon	5.17	273.34	290.70	727.63	722.56	320.59	6.54
Avg.	5.55	305.49	293.30	686.94	676.05	338.10	7.58

Table 4. The average computation time (sec) for various vector quantization algorithms (bit rate =0.3750)

Codebook size: 64							
Image	LBG	PSO-LBG	QPSO-LBG	HBMO-LBG	FF-LBG	BA-LBG	Proposed
Lena	4.95	312.44	322.11	746.34	712.45	322.12	6.35
Pepper	5.76	304.03	307.95	637.96	632.62	315.60	7.18
Baboon	6.72	315.85	623.92	776.23	764.08	396.51	8.10
Avg.	5.81	310.77	417.99	720.17	703.05	344.74	7.21

Table 5. The average computation time (sec) for various vector quantization algorithms (bit rate = 0.4375)

Codebook size: 128							
Image	LBG	PSO-LBG	QPSO-LBG	HBMO-LBG	FF-LBG	BA-LBG	Proposed
Lena	11.88	521.28	835.05	888.32	868.07	516.17	13.28
Pepper	16.42	508.58	571.99	956.73	915.46	528.30	17.81
Baboon	19.62	406.38	466.01	965.67	921.12	838.09	21.01
Avg.	15.97	478.74	624.35	936.90	901.55	627.52	17.36

Table 6. The average computation time (sec) for various vector quantization algorithms (bit rate = 0.5000)

Codebook size: 256							
Image	LBG	PSO-LBG	QPSO-LBG	HBMO-LBG	FF-LBG	BA-LBG	Proposed
Lena	20.91	877.94	895.82	798.35	788.13	666.04	21.28
Pepper	18.11	752.58	751.77	998.98	973.20	568.21	19.51
Baboon	28.02	594.62	563.98	1012.56	1031.44	568.64	29.41
Avg.	22.34	741.71	737.19	936.76	930.92	600.96	23.04

Table 7. The average computation time (sec) for various vector quantization algorithms (bit rate = 0.5625)

Codebook size: 512							
Image	LBG	PSO-LBG	QPSO-LBG	HBMO-LBG	FF-LBG	BA-LBG	Proposed
Lena	56.31	1157.90	1197.31	1791.44	1716.80	1342.51	57.72
Pepper	82.00	1751.23	1852.80	1388.90	1357.86	1114.21	83.39
Baboon	63.43	2011.19	2186.58	2179.56	2212.43	2459.56	64.69
Avg.	67.24	1640.10	1745.56	1786.63	1762.36	1638.76	68.06

5. Conclusion

Based on kernel density estimation, an efficient method for VQ-codebook generation has been proposed. In terms of computation time (sec) and PSNR, the proposed algorithm outperforms LBG, PSO-LBG, QPSO-LBG, HBMO-LBG, and FA- LBG. Based on the simulation results, the proposed method is faster than that of the remaining six algorithms. Furthermore, the reconstructed images improve in quality.

References

[1] *Hagag A, Amin M and Abd El-Samie F. E.: "Multispectral image compression with band ordering and wavelet transforms", Signal, Image and Video Processing, vol.9, no. 4,* ***(2015)*** *pp. 769–778.*

[2] *Bruylants T, Munteanu A and Schelkens P, "Wavelet based volumetric medical image compression", Signal Processing: Image Communication, vol. 31,* ***(2015)*** *, pp. 112-133.*

[3] *Ma X, Pan Z and Li Y, "High-quality initial codebook design method of vector quantisation using grouping strategy", IET Image Processing, vol. 9, no.11,* ***(2015)****, pp. 986 – 992.*

[4] *Thakur V. S, Gupta S and Thakur K, "Hybrid WPT-BDCT transform for high-quality image compression", IET Image Processing, vol.11, no.10,* ***(2017)****, pp. 899 – 909.*

[5] *Sharma A and Sundaram S, "An enhanced contextual DTW based system for online signature verification using Vector Quantization", Pattern Recognition Letters, 2016, vol. 84,* ***(2016)****, pp. 22-28.*

[6] *Feng, Ya-Pei; Lu, Zhe-Ming; Li, Hui, "Image coding based on classified vector quantization using edge orientation patterns", IET Image Processing, vol.11, no.10,* ***(2017)****, pp. 910 – 918.*

[7] *Saleem M, Rehman Z. U and Zahoor U, "Self learning speech recognition model using vector quantization", Sixth International Conference on Innovative Computing Technology (INTECH), Dublin, Ireland,* ***(2016).***

[8] *Rahmani P and Dastghaibyfard G, "Two reversible data hiding schemes for VQ-compressed images based on index coding", IET Image Processing, vol.12, no.7,* ***(2018)****, pp. 1195 – 1203.*

[9] *Pan Z, Ma L and Deng M, "New reversible full embeddable information hiding method for vector quantisation indices based on locally adaptive complete coding list", IET Image Processing, vol. 9, no.1,* ***(2015)****, pp. 22 – 30.*

[10] *Tiwari A, Sharma M and Tamrakar R. K, "Watermarking based image authentication and tamper detection algorithm using vector quantization approach", AEU - International Journal of Electronics and Communications, vol.78,* ***(2017)****, pp. 114-123.*

[11] *Linde Y, Buzo A and Gray R. M, "An algorithm for vector quantizer design", IEEE Transactions on Communications, vol.28, no.1,* ***(1980)****, pp. 84–95.*

[12] *Chen Q, Yang J.G and Gou J, "Image compression method using improved PSO vector quantization", First international conference on neural computation (ICNC), Changsha, China,* ***(2005)*** *, pp. 490 – 495.*

[13] *Wang Y, Feng X. Y, Huang X. Y, et al., "A novel quantum swarm evolutionary algorithm and its applications", Neuro computing, vol.70, no.(4-6),* ***(2007)****, pp. 633-640.*

[14] *Horng M. H and Jiang T.W, "Image vector quantization algorithm via honey bee mating optimization", Expert Systems with Applications, vol.38, no.3,* ***(2011)****, pp. 1382-1392.*

[15] *Horng M. H, "Vector quantization using the firefly algorithm for image compression", Expert Systems with Applications, vol.39, no.1,* ***(2012)****, pp. 1078-1091.*

[16] *C. Karri, C. Jena U, "Fast vector quantization using a Bat algorithm for image compression", Engineering Science and Technology, an International Journal, vol.19, no.2,* ***(2016)****, pp. 769-781.*

[17] *Darroudi A, Parchami J, Razavi M. K, et al., "EEG adaptive noise cancellation using information theoretic approach", Bio-Medical Materials and Engineering, 2017, vol. 28, no. 4,* ***(2017)****, pp. 325-338.*

[18] *Pratibha Pramod Chavan, B Sheela Rani, M Murugan and Pramod Chavan, "A novel image compression model by adaptive vector quantization: Modified rider optimization algorithm", Sadhana, vol.45, no.232,* ***(2020)****, pp. 1-15.*

[19] *V. R Kavitha, M. Kanchana, B. Gobinathan, K. R. Sekar and Mohamed Yacin Sikkandar, "Optimization Based Vector Quantization for Data Reduction in Multimedia Applications", Intelligent Automation & Soft Computing, vol.31, no.2,* ***(2021)****, pp. 853-867.*