Volume 3, Issue 6, June 2013 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

Image Compression Using Vector Quantization Algorithms: A Review

Mukesh Mittal, Ruchika Lamba
Department of Electrical &
Instrumentation Engineering
Thapar University, Patiala, India

Abstract: This paper presents a review of vector quantization algorithmsused for the designing of the codebook to obtain better quality of the image with minimum distortion. First, the concept of vector quantization is introduced and then different algorithms such as Linde-Buzo-Gray (LBG) algorithm, Equitz nearest neighbor (ENN) algorithm,Back propagation neural network(BPNN) algorithm and fast back propagation(FBP) algorithm are reviewed.

KeyWords: Image Compression, LBG, ENN, BPNN, FBP.

I. INTRODUCTION

Image compression is essential for applications such as TVtransmission, video conferencing, facsimile transmission of printed material,graphics images[1].A fundamental goal of image compression is to reduce the bit rate for transmission or data storage while maintaining an acceptable fidelity or image quality.Every digital image is specified by the number of pixels associated with the image.Each pixel in an image can be denoted as a coefficient, which represents the intensity of the image at that point.Once compressed, the coded image is transferred to the receiving end, where these compressed images are again decompressed to recover the original image[2].

II.VECTOR QUANTIZATION

The most powerful and quantization technique used for the image compression is vector quantization(VQ).The vector quantization algorithms for reducing the transmission bit rate or storage have been extensively investigated for speech and image signals.Image vector quantization (VQ) includes four stages: vector formation,Training set selection, codebook generation and quantization. The first step is to divide the input image into set of vectors.TheSubset of vectors in the set is later chosen as a training sequence. The codebook of codewords is obtained by an iterative clustering algorithm. Finally, in quantizing an input vector, closest codewords in the codebook is determined and corresponding label of this code word is transmitted. In this process, data compression is achieved because address transmission requires fewer bits than transmitting vector itself. The concept of data quantization is extended from scalar to vector data of arbitrary dimension. Instead of output levels, vector quantization employs a set of representation vectors (for one dimensional case) or matrices (for two dimensional cases). Set is defined as "codebook" and entries as "codewords". Vector quantization has been found to be an efficient coding technique due to its inherent ability to exploit the high correlation between the neighboring pixels[3].

III. LBG ALGORITHM

Generalized Lloyd Algorithm(GLA), which is also called,Linde-Buzo-Gray (LBG)AlgorithmThey used a mappingfunction to partition training vectors into N clusters. The mapping function is defined as:
$R^k \rightarrow CB$
Let $X = (x_1, x_2 \ldots x_k)$ be a training vector and d(X; Y) be the EuclideanDistance between any two vectors. The iteration of GLA for a codebook generation isgiven as follows:
Step 1: Randomly generate an initial codebook CB_0.
Step 2: i = 0.
Step 3: Perform the following process for each training vector.
Compute the Euclidean distances between the training vector and the codewords in CB_i. The Euclidean distance is defined as
$d(X; C) = (\sqrt{\sum_{t=1}^{k}}(x_t - c_t)^2)$…….. (1)
Search the nearest codeword among CB_i.
Step 4:Partition the codebook into N cells.
Step 5**:** Compute the centroid of each cell to obtain the new codebook CB_{i+1}.
Step 6**:** Compute the average distortion for CB_{i+1}. If it is changed by a smallenough amountsince the last iteration, thecodebook may converge and the procedure stops. Otherwise, i = i + 1 and go to Step 3.

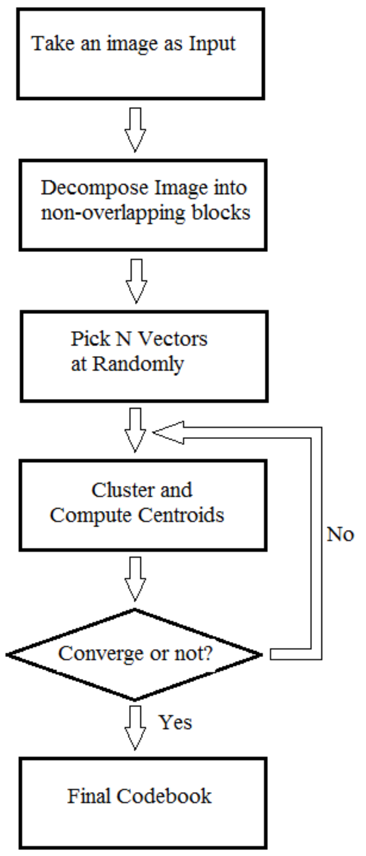
Take an image as Input
Decompose Image into non-overlapping blocks
Pick N Vectors at Randomly
Cluster and Compute Centroids
No
Converge or not?
Yes
Final Codebook

Fig 1.Flowchart of LBG algorithm [4]

LBG algorithm has the local optimal problem and the utility of each codeword in the codebook is low. The local optimal problem means that codebook guarantees local minimum distortion but not global minimum distortion[5].

IV.EQUITZ NEAREST NEIGHBOUR (ENN)

The selection of initial codebook by LBG algorithm is poor which results in an undesirable final codebook. Another algorithm Equitz Nearest neighbor (ENN) is used in which no need for selection of initial codebook. At the beginning of ENN algorithm, all training vectors are viewed as initial clusters (codevectors). Then, the two nearest vectors are found and merged by taking their average. A new vector is formed which replaces and reduce the number of clusters by one. Process is going on until desired number of clusters is not obtained[6].
The steps for the implementation of the ENN algorithm are as follows:
1. Initially, all the image vectors taken as the initial codewords.
2. Find each of two nearest codewords by the equation:
$Od(X, Yi)={}^{k-1}\sum_{j=0}|X_j–Y_{i,j}|$……… (2)
Where 'X' represents an input vector from the original image and 'Y' represents a codeword, and merge them by taking their average where 'k' represents the codeword length.
3. The new codeword replaces the two codeword and reduce the number of codewords by one.
4. Repeat step 2 and step 3 until desired number of codewords is reached.
The ENN requires long time and large number of iterations to design the codebook. Therefore, to decrease the number of iterations and time required to generate the codebook, an image block distortion threshold value(dth)is calculated.

The previous algorithm is modified as:
1. Determine the desired number of codeword and the maximum number of gray levels in the image (max–gray)
2. Distortion threshold (dth) is calculated as:
$dth=k\times(max\text{-}gray/64)$………. (3)
Where 'k' is codeword length
3. Calculate the distortion error between a taken codeword and a next codeword. If distortion error is less than or equal to dth, then merge these two codeword and reduce the number of codeword by one. Otherwise, consider the next codeword.
4. Repeat the step 3 until we obtain number of codewords equal to desired number of codewords.
5. Even, after all the codewords are compared and merged, the resultant number of codewords greater than desired number of codewords, then change the dth value as follows
$Dth=dth+k\times(max\text{-}gray/256)$……………….. (4)
And then go to step 3.

V.BACK PROPAGATION NEURALNETWORK (BPNN) ALGORITHM

BPNN algorithm helps to increase the performance of the system and to decrease the convergence time for the training of the neural network[7]. BPNN architecture is used for both image compression and also for improving VQ of images. A BPNN consists of three layers: input layer, hidden layer and output layer. The number of neurons in the input layer is equal to the number of neurons in the output layer. The number of neurons in the hidden layer should be less than that of the number of neurons in the input layer. Input layer neurons represent the original image block pixels and output layer neuron represents the pixels of reconstructed image block. The assumption in hidden layer neurons is that the arrangement is in one-dimensional array of neurons which represents the element of codeword. This process produces an optimal VQ codebook. The source image is divided into non-overlapping blocks of pixels such that block size equals the number of input layer neurons and the number of hidden layer neurons equals the codeword length. In the BP algorithm, to design the codebook, the codebook is divided into rows and columns in which rows represent the number of patterns of all images and columns represents the number of hidden layer units.

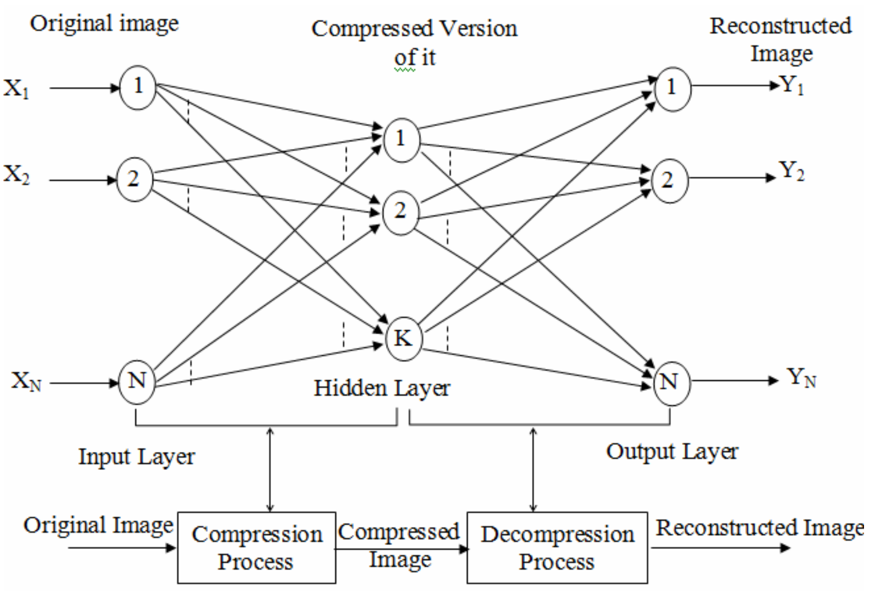

Fig 2 BPNN image compression system[9]

The implementation of BPNN VQ encoding can be summarized as follows:
1.Divide source image into non-overlapping blocks with predefined block dimension(P), where (P×P) equals the number of neurons in the input layer.
2. Take one block from the image, normalize it, convert the image into pixels (rasterizing), and apply it to the input layer neurons of BPNN.
3. Execute one iteration BP in the forward direction to calculate the output of hidden layer neurons.
4. From the codebook file, find the codeword that best matches the outputs of hidden layer neurons.
5. Store index i.e. position of this codeword in codebook in the compressed version of source image file.
6. For all the blocks of source image, repeat the steps from step2 to step5.
Number of bits required for indexing each block equals to $\log_2 M$, where M is codebook length
The implementation of BPNN VQ decoding process can be described as follows:
1. Open compressed VQ file
2. Take one index from this file.
3. This index is then replaced by its corresponding codeword which is obtained from the codebook and this codeword is assumed to be the output of hidden layer neurons.
4. Execute one iteration BP in the forward direction to calculate the output of the output layer neurons, then de-rasterizing it,de-normalize it and store this output vector in a decompressed image file.
5. Repeats steps from step2 to step4 until the end of compressed file.
The BP algorithm is used to train the BPNN network to obtain the codebook with smaller size with improved performance of the system. BPNN image compression system has the ability to decrease the errors that occurs during transmission of compressed images through analog or digital channel. Practically, we can note that BPNN has the ability to enhance any noisy compressed image that has been corrupted during compressed image transmission through a noisy digital or analog channel. BPNN has the capacity to compress untrained images but not in the same performance of trained images. This can be done especially when using small number of image block dimension[8].

VI. FAST BACK PROPAGATION (FBP) ALGORITHM

The FBP algorithm is used for training the designed BPNN to reduce the convergence time of BPNN as possible as. The fast back propagation(FBP) algorithm is based on minimization of objective function after initial adaption cycles. This minimization can be obtained by reducing lambda (λ) from unity to zero during network training. The FBP algorithm is differ from standard BP algorithm in the development of alternative training criterion. This criterion indicates that (λ)must change from 1 to 0 during training process i.e. λ approaches to zero as total error decreases. In each adaption cycle, λ should be calculated from the total error at that point, according to the equation: $\lambda=\lambda(E)$, where E is error of network, indicates that $\lambda\approx1$ when E>>1.When E>>1 for any positive integer n, $1/E^n$ approaches zero, therefore $\exp(-1/E^n)\approx1$. When E<<1, $1/E^n$ is very large, therefore $\exp(-1/E^n)\approx0$. As a result, for the reduction of λ from 1 to 0, a suitable rule is as follows[9]:

$\lambda=\lambda(E)=\exp(-\mu/E^n)$…………(5)

Where μ is positive real number and n is positive integer. When n is small, reduction of λ is faster, when E>>1. It has been experimentally verified that if λ is much smaller than unity during initial adaption cycles,algorithm may be trapped in local minimum. So, n should be greater than 1.

Thus, λ is calculated during any network training according to equ. 6[9]:

$\lambda = \lambda(E)=\exp(-\mu/E^2)$………. (6)

In FBP algorithm, all the hidden layer neurons and output layer neurons uses hyperbolic tangent function instead of sigmoid functions in the BPNN architecture. So, equation is modified for hyperbolic tangent function as follows[9]:

$F(NET_j)= (e^{NET_j} - e^{-NET_j}) / (e^{NET_j} + e^{-NET_j})$……. (7)

And derivative of this function is as follows:

$F'(NET_j)=(1-(F(NET_j))^2)$…………(8)

So that $F(NET_j)$ lies between -1 and 1.

VII. CONCLUSION

From all the algorithms mentioned above we have concluded that the fast back propagation algorithm (FBP) is best of all the four algorithms based on the parameters mentioned in table 1. FBP trains the BPNN image compression system to speed up the learning process and reduce the convergence time. The performance of FBP is better of all the algorithms and can be increased by modifying the network itself i.e. changing number of input layer neurons and hidden layer neurons.

TABLE 1COMPARISON OF VARIOUS ALGORTHMS OF VECTOR QUANTIZATION

PARAMETERS	**LBG ALGORITHM**	**ENN ALGORITHM**	**BPNN ALGORITHM**	**FBP ALGORITHM**
CODE BOOK SIZE	This algorithm designs very large super codebook	This algorithm designs small codebook as compared to LBG algorithm.	This algorithm designs codebook with smaller size as compared to ENN algorithm	Codebook size is same as that of BPNN algorithm.
CODE WORD SIZE	The size of each codeword in the codebook is P×P, where P is the dimension of image block.	The size of codeword in the codebook is P×P, where P is the dimension of image block	The size of each codeword in codebook is equal to number of hidden layer neurons.	. Size of each codeword is same as BPNN algorithm
STORAGE SPACE	It requires more storage space for the codebook.	It requires less storage space for the codebook	It requires less storage space as compared to ENN algorithm.	It requires same storage space as that of BPNN algorithm
CODEBOOK GENERATION TIME	It takes long time for the generation of the codebook.	It takes less time for the generation of the codebook.	. It takes less time for the generation of the codebook as compare to ENN	It takes less time for the generation of codebook than BPNN algorithm
COMPUTATIONAL COMPLEXITY	A complete design requires a large number of computations	This algorithm reduces the computations dramatically	Computational load is less for encoding and decoding process as compared to ENN algorithm	Computational load is less for encoding and decoding process as compared to ENN algorithm
CONVERGENCE TIME	Convergence time is very large	Convergence time is less than LBG algorithm	Convergence time is less than that of the ENN algorithm.	FBP trains the BPNN image compression system to speed up the learning process and

				reduce the convergence time
PERFORMANCE	Performance of this algorithm is not so good.	. Performance is better than LBG algorithm as LBG selects the initial codebook randomly	Performance is far much better than ENN algorithm	Performance is better than BPNN algorithm

REFERENCES:

[1] Image coding using vector quantization: A review by Nasser.N. Nasrabadi, Member, IEEE and Robert. A.King IEEE, Transactions on Communications, VOL. 36, NO. 8, AUGUST 1988.

[2] Developing neural networks applications using labview thesis by Pogula Sridhar, Sriram and Dr. Robert W. McLaren, Thesis Supervisor, July 2005.

[3] A multilayer addresses vector quantization technique by Nasser. N. Nasrabadi, Member, IEEE and Yushu Feng, student,Member,IEEE.IEEE Transactions on Circuits and Systems, VOL37, NO7, July 1990

[4] An efficient codebook initialization approach for LBG algorithm by Arup Kumar Pal Department of Computer Science and Engineering, NIT Jamshedpur, India and Anup Sar Department of Electronics and Telecommunication Engineering, Jadavpur University, India.International Journal of Computer Science, Engineering and Applications (IJCSEA) Vol.1, No.4, August 2011.

[5] A survey of VQ codebook generation by Tzu-Chuen-Lu Ching-Yun Chang.Journal of Information Hiding and Multimedia Signal Processing @2010 ISSN 2073-4212, Ubiquitous InternationalVolume 1, Number 3, July 2010

[6] Codebook enhancement in vector quantization in image compression using back propagation neural network by Omaima N.A. AL-Allaf.Journal of applied sciences 11(17): 3152-3160,2011 ISNN 1812-5654/DOI:10.3923/jas.2011 31523160, 2011 Asian network for scientific information.

[7] Image compression using back propagation neural network by S.S.Panda, M.S.R.S. Prasad, MNM Prasad,ChSKVR Naidu. [IJESAT] INTERNATIONAL JOURNAL OF ENGINEERING SCIENCE & ADVANCED TECHNOLOGY, Volume - 2, ISSN: 2250–3676 Issue - 1, 74 – 78.

[8] Improving the performance of back propagation neural network algorithm for image compression/decompression system by Omaima N.A. AL-Allaf. Journal of Computer Science 6 (11): 1347-1354, 2010 ISSN 1549-3636© 2010 Science Publications.

[9] Fast back propagation neural network algorithm for reducing convergence time of BPNN by Omaima N.A. AL-Allaf.Proceedings of the 5th International Conference on IT & Multimedia at UNITEN (ICIMU 2011) Malaysia