

## Compression of Image Using VHDL Simulation

1) **Prof. S. S. Mungona** (Assistant Professor, Sipna COET, Amravati). 2) **Vishal V. Rathi**,

### **Abstract :**

Maintenance of all essential information without any deletion of data or important information and yet preparing a compact representation of an Image is technically referred to as Image compression. Goal of an Image compression is to reduce image with minimum number of bits of a acceptable image quality. Compression is achieved using redundancy. Here, redundancy means duplication.

There are basically two schemes of image compression :

- 1) Lossless Image Compression scheme.
- 2) Lossy Image compression scheme.

High compression can be easily achieved using lossy than lossless image compression scheme. Image compression can be easily achieved using LBG algorithm.

Linde, Buzo and Gray algorithm is most cited and widely used algorithm on designing of the VQ codebook. This is an algorithm developed in the community of Vector quantization (VQ) for the purpose of data compression.

However, the performance of the standard LBG algorithm is highly dependent on the choice of the initial codebook. A training set of Images is generally used for generation of codebook. Generated codebook is stored into text file for Vhdl file handling or data array in Vhdl code.

An initial codevector is set as average of entire training sequence is later on split to provide 2

codevectors. These are further split to double themselves and the process is repeated to procure the desired number.

This compression algorithm is evaluated using compression ratio (CR) and Peak signal to Noise Ratio (PSNR) using this method.

Reduction can be achieved by avoiding the computation of unnecessary code words. This will thus help in reducing computation cost and provides a flexible way of selection for test condition to accommodate to different training set of Images.

**Keywords:** Image compression, LBG algorithm, Vector quantization, image compression schemes.

### **Introduction :**

Fundamental goal of image compression is to reduce bit rate for transmission or data storage while maintaining an acceptable image quality[2]. Maintenance of all the essential information without any deletion of data or important information and yet preparing a compact representation of an image is technically referred to as image compression.

In this project, we are implementing the LBG algorithm for image compression. One of the important factors for image storage or transmission over any communication media is the image compression. Compression makes it possible for creating file sizes of manageable, storable and transmittable dimensions. Compression is achieved by exploiting the redundancy. Image Compression techniques fall under two categories, namely: *Lossless and Lossy*.

In *Lossless* techniques the image can be reconstructed after compression, without any loss of data in the entire process. The image compression and decompression is identical to original image and every bit of information is preserved under decomposition process. Reconstructed image is replica of original image. Although there is deterioration in image quality. Lossless image compression is basically used in those where no loss of data can be compromised. It can be used for document and medical imaging. *Lossy* techniques, on the other hand are irreversible, because, they involve performing Quantization, which result in loss of data. *Lossy compression* can be used for signals like natural images, speech [2]etc, where the amount of loss in the data determines the quality of the reconstruction and does not lead to change in the information content. Reconstructed image contains degradation with respect to its original image. There are small amount of redundancies present. It can be used for multimedia applications.

More compression is achieved in the case of lossy compression than lossless compression. Compression of images involves taking advantage of the redundancy in data present within an image. Vector quantization is often used when high compression ratio are required. Any compression algorithm is acceptable provided a corresponding decompression algorithm exists. Vector Quantization(VQ) achieves more compression than scalar quantization, making it useful for band-limited channels. Numerous

compression techniques have been developed such as vector quantization, block truncation method, transform coding, hybrid coding and various adaptive versions of this methods. Among these techniques, vector Quantization is widely used in image compression owing to its simple structure and low bit rate. In the process of vector Quantization the image to

be encoded is segmented into a set of input image vectors. The most important task for the VQ scheme is to design a good codebook. A good codebook is required because the reconstructed image highly depends on the codewords in this very codebook. The generated codebook store into text file for vhdl file handling or data array in vhdl code.

The algorithm for the design of optimal VQ is commonly referred to as the Linde-Buzo-Gray(LBG) algorithm, and it is based on minimization of the squared-error distortion measure.

In 1980, Linde, Buzo, and Gray proposed the VQ scheme for grayscale image compression and it has proven to be a powerful tool for both speech and digital image compression. There are three major procedures in VQ, namely codebook generation, encoding procedure and decoding procedure. In the codebook generation process, various images are divided into several k-dimension training vectors. The representative codebook is generated from these training vectors by the clustering techniques. In the encoding procedure, an original image is divided into several k-dimension vectors and each vector is encoded by the index of codeword by a table look-up method. The encoded results are called an index table [6].

#### **System Architecture :-**

The two basic tasks of vector quantization image compression are the design of the codebook and search for the best approximation (codeword) for each block. The most popular technique is the LBG. The key point in the codebook design procedure is the design of the good initial codebook which will require a small number of optimization iterations and will lead to final codebook. We can view blocks as vectors, hence the name vector quantization.

Vector Quantization (VQ) is a lossy data compression method based on the principle of Block

Coding. It is a fixed- to- fixed length algorithm .The design of the vector quantizer (VQ) is considered to be a challenging problem due to the need for multi-dimensional integration .[3] Linde, Buzo and Gray (LBG) proposed a VQ design algorithm based on a *training sequence*. The use of the training sequence bypasses the need for *multidimensional integration* .[1]

Each quantizer codeword represents a single sample of the source output. By taking longer sequences of input samples, it is possible to extract the structure in the source coder output. Even when the input is random, encoding sequences of samples instead of encoding individual samples separately provides a more efficient code. Encoding sequences of samples is more advantageous in lossy compression framework as well. By advantageous we mean a lower distortion for a given rate or a lower rate for a given distortion. By rate we mean the average number of bits per input samples and the measure of distortion will generally be the mean squared error and the peak signal to noise ratio. The idea that encoding sequences of outputs can provide an advantage over the encoding of individual samples and the basic results in information theory were all proved by taking longer and longer sequences of input [4]. The most prevalent technique for codebook design is the generalized Lloyd algorithm (GLA). Initially, Lloyd developed an algorithm for scalar quantizer design. The algorithm was later generalized by Linde et al. for used in vector quantization. The GLA is therefore sometimes referred to as the LBG algorithm. The LBG algorithm also bears close resemblance to the K- mean algorithm used in data clustering. Hence, it is sometimes referred to as the cluster compression algorithm[3].

The LBG algorithm is the most cited and widely used algorithm on designing the VQ codebook. It is the starting point for most of the work on vector

quantization. The performance of the LBG algorithm is extremely dependent on the selection of the initial codebook. In conventional LBG algorithm, the initial codebook is chosen at random from the training data set. It is observed that some-time it produces poor quality codebook. Due to the bad codebook initialization, it always converges to the nearest local minimum. This problem is called the local optimal problem . In addition, it is observed that the time required to complete the iterations depends upon how good the initial codebook is. In literature, several initialization techniques have been reported for obtaining a better local minimum. The concept of VQ is based on Shannon"s rate-distortion theory where it says that the better compression is always achievable by encoding sequences of input samples rather than the input samples one by one. In VQ based image compression, initially image is decomposed into non-over lap-ping sub image blocks. Each sub block is then converted into one-dimension vector which is termed as training vector. From all these training vectors, a set of representative vectors are selected to represent the entire set of training vectors [5]. As stated earlier, the VQ process is done in the following three steps namely (i) codebook design, (ii) encoding process and (iii) decoding process. An initial codevector is set as average of entire training sequence is later on split to provide 2 codevectors. These are further split to double themselves and the process is repeated to procure the desired number.

An identical codebook previously generated is required in both the encoding procedure and the decoding procedure in VQ scheme. In the process of vector quantization, the image to be encoded is segmented into set of input image vectors. In the encoding procedure, the closest codeword for each input vector is chosen, and its index is transmitted to

the receiver. In the decoding procedure, a simple table look-up procedure is done to reconstruct the encoded image in the receiver. Thus, the encoded image of the original input image becomes available to the receiver. The whole compression process is accomplished when the encoded image is reconstructed with the corresponding index of each input image vector.

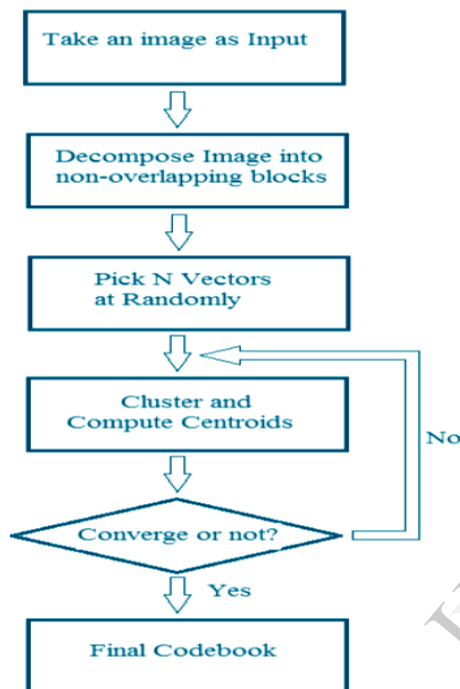


Figure 1 : The flowchart of LBG clustering algorithm.

The flowchart of LBG clustering algorithm is shown in Figure 1. After codebook design process, each codeword of the codebook is assigned a unique index value. Then in the encoding process, any arbitrary vector corresponding to a block from the image under consideration is replaced by the index of the most appropriate representative codeword. The matching is done based on the computation of minimum squared Euclidean distance between the input training vector and the codeword from the codebook. So after encoding process, an index table is produced. The codebook and the index-table is nothing but the compressed form of the input image. In decoding process, the codebook which is available at the receiver

end too, is employed to translate the index back to its corresponding codeword. This decoding process is simple and straight forward. shows the schematic diagram of VQ encoding-decoding process [5]. LBG is an easy and rapid algorithm. However, it has the local optimal problem which is that for a given initial solution, it always converges to the nearest local minimum. In other words, LBG is a local optimization procedure [6].

The objective of our project is to compress the image using the Linde, Buzo, and Gray (LBG) Algorithm.

The algorithm can be explained using few simple steps :-

**Step 1.** First, you find the sample mean  $z_1^{(1)}$  for the entire data set. Here we have only one prototype. The sample mean is proven total mean square distortion. for a single prototype.

**Step 2.** Set  $k = 1, l = 1$ .  $l$  is the index for the iteration.  $k$  counts the number of prototypes that have been generated. Here we have only one prototype.

**Step 3.** If  $k < M$ , split the current centroids by adding small offsets.

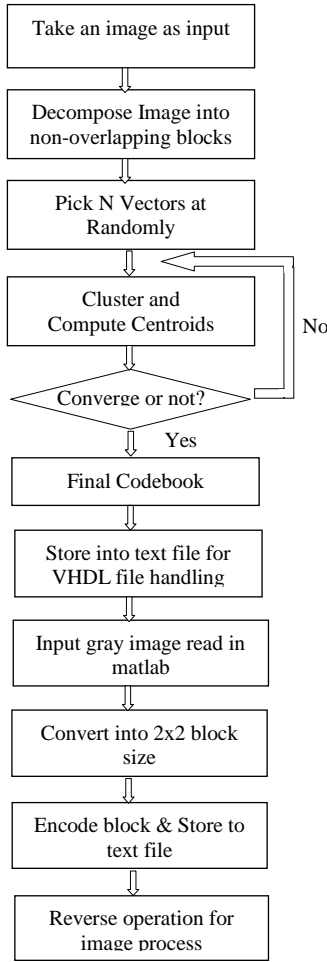
Since if we already have  $k$  prototypes, we need  $M - k$  additional prototypes.

If  $M - K \geq K$ , Split all the existing centroids that have been created so far; otherwise we split only  $M - K$  of them.

**Step 4.** For example, to split  $z_1^{(1)}$  into two centroids, let  $z_1^{(2)} = z_1^{(1)}, z_2^{(2)} = z_1^{(1)} + \epsilon$ , where  $\epsilon$  is a small offset.

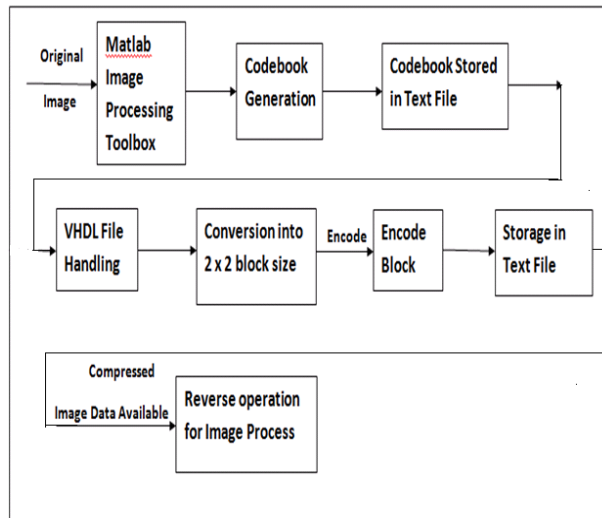
**Step 5.** Use  $\{z_1^{(1)}, z_2^{(1)}, \dots, z_k^{(1)}\}$  as initial prototypes, which includes the previously generated centroids and the newly split centroids.

**Step 6.** Check whether the number of prototypes has reached the target number of prototypes. In other words, if  $k < M$ , go back to step 3; otherwise, stop.



**Block Diagram:**

**Description**



:

The algorithm requires an initial codebook to start with. We can use matlab image processing tool

box for Codebook generation. Generated Codebook store into text file for vhdL file handling or data array in vhdL code Gray input image is read in matlab and store in to text file for vhdL file handling or data array in vhdL code.

We have to convert the whole image into 2x2 block size. Encode / convert each block to value for which codebook use. Store in text file / send these encode values as compress image data

For restore process we process reverses operation.

And this algorithm works only on gray image, and code nature will be non-synthesis as it use file handling / input is image. The algorithm can be evaluated in terms of compression ratio (CR) and peak signal to noise ratio (PSNR).

The compression ratio and peak signal to noise ratio are defined along with their mathematical formula and are given as under:-

Compression ratio is defined as ratio of the number of bits required to represent the data before compression to the number of bits required after compression. (10)

Mathematically,

Compression ratio (%) =

$$\frac{\text{No of bits required before compression}}{\text{No of bits required after compression}}$$

Peak signal to noise ratio(PSNR) is defined as ratio of square the peak value of the signal to the mean square error.

Where, mean square error refers to the average value of the square of the error between the original image f(m,n) and the reconstruction image g(m,n). A common measure of distortion is the mean square error (MSE). (10)

Mathematicay,

$$MSE = \frac{1}{M \times N} \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} (f(m,n) - g(m,n))^2$$

$M \times N$  represents the size of the image.

The distortion in the decoded images is measured using peak signal to noise ratio, (10)

$$\text{PSNR} = 10 \log_{10} \left( \frac{255^2}{\text{MSE}} \right) \text{ dB}$$

#### Applications :

The image compression has diverse application including:

- 1) Document and medical imaging
- 2) Remote sensing
- 3) Tele-video conferencing

#### Conclusion :

Data transfer of uncompressed image over digital networks require very high bandwidth. The state of the art image compression techniques may exploit the dependencies between the subbands is a transferred image. The above proposed algorithm reduces the complexity of a transferred image without sacrificing performance. Vector quantization is an established lossy compression technique that has been used successfully to compress signals such as speech, music, video, imagery.

#### Future Scope:

LBG is an easy and Rapid algorithm. However, it has the local optimal problem which is that for a given initial solution, it always converges to the nearest local minimum. In other words, LBG is a local optimization procedure. The other problem with LBG algorithm is the codeword generation process needs a great deal of calculation. Thus, LBG is relatively slow algorithm.

Consider the shown example of a squirrel. The original image has undergone compression and its ratio, along with mean square error (MSE) and peak signal to noise ratio are elaborately compared.

Original Image



Reconstructed Image



Table Compression measure are shown					
Image	Codebook Size	Bits Needed	CR	MSE	PSNR
Squirrel	128	7	4:1	468.24	21.42

#### References :

- [1] P.Franti, T.Kaukoranta, D.-F.Shen and K.-S.Chang, "Fast and memory efficient implementation of the exact PNN", IEEE Transactions on Image Processing, 9 (5), 773-777, May 2000.
- [2] Chin-Chen Chang and Yu-Chen-Hu, "Fast LBG codebook Training algorithm for vector quantization". 0098-3063/98/IEEE.
- [3] Peter Veprek, A.B.Bradley, "An Improved Algorithm for vector quantizer design" 1070-9908 IEEE.
- [4] Ms. Asmita A. Bardekar, Mr. P.A Tijare, "Implementation of LBG algorithm for image compression". International Journal of Computer Trends and Technology- Volume 2 Issue 2-2011.
- [5] Arup Kumar Pal and Anupkar, "An Efficient codebook Initialization Approach for LBG algorithmic International Journal of Computer Science, Engineering and Application" Vol.1. No:4 August 2011.
- [6] Manoj Kumar, Poonam Saini, "Image Compression with Efficient Codebook initialization using LBG Algorithm" ISSN: 2319-7900, IJACT.
- [7] Momotaz Begum, Nurun Nahar, Kaneez Fatimah and Md. Kamrul Hasan, "A New Initialization Technique for LBG Algorithm", 2<sup>nd</sup> International Conference on Electrical

and Computer Engineering ICECE 2002, Bangladesh, pp-26-28,2002.<

[8] Ming Yang and Nikolaos Bourbakis,

“An Overview of Lossless Digital Image Compression Techniques,” 48<sup>th</sup> Midwest Symposium on Circuits and Systems, Vol. 2, pp. 1099-1102, 2005.

[9] N Akrouf, R Prost and R Goutte,

“Image Compression by vector Quantization: a review focused on codebook generation,” Image and Vision Computing, Vol. 12, No. 10, pp.627-637, Dec. 1994.

[10] “Digital image processing” by S. Jayaraman, S. Esakkirajan, T. Veerakumar – Tata Mc Graw Hill.

IJERT