Volume 4, Issue 7, July 2014 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

# Image Compression through VHDL Simulation

[1]Prof. S. S. Mungona, [2]Mr. Vishal V. Rathi
[1]Sipna College of engineering & Technology, India
[2]M.E. II[nd] year (Digital Electronics) Sipna College of engineering & Technology, India

***Abstract— The ever rising usage of multimedia and other such modern techniques are responsible for the generation of large amount of data. This data requires large amount of space for storage in order to solve this problem for image compression is an efficient technique for which reduces storage space without loss in data. Compression helps in sending and receiving large files at high transfer rate. For image compression, we use linde, Buzo, and gray algorithm for image compression. LBG algorithm is also called as cluster compression algorithm. The basic requirement of LBG algorithm is the generation of codebook. codebook is the collection of several codewords.***
***The algorithm can evaluated using compression ratio, mean square error. The distortion in the decoded image can be determined using peak signal to noise ratio.***

***Keywords— multimedia, compression, LBG algorithm, codebook, peak signal to noise ratio.***

## I. INTRODUCTION

Image compression have a wide area of applications like in televideoconferencing, document and medical image and remote sensing [5]. Main goal of image compression is to reduce bit rate and maintaining acceptable image quality.

Image compression can be achieved using two image compression schemes defined as below :-

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

Lossy Image compression scheme is further classified as :-

1) Scalar Quantization (SQ)
2) Vector Quantization (VQ)

Lossless image compression scheme can be defined as a scheme where the image can be reconstructed after compression, without 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.

In this scheme, 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 [1]etc, vector quantization achieves more compression than scalar quantization, thus making it useful band-limited channels. Vector quantization is widely used because of its low bit rate.

Vector quantization is a classical technique from signal processing and image compression which allow the modelling 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 bits (vector) into groups having approximately the same number of points closest to them. Each group is represented by its centroid point, as in k-means and some other algorithms. [8]

The density matching property of vector quantization is powerful, especially for identifying the density of large and high dimensional data. Since data points are represented by the index of their closest centroid, commonly occurring data have low error. and rare data have high error. That's why, VQ is suitable for lossy data compression. It can be used for lossy data correction and density estimation.[8]

Principles of vector quantization is to assist a project of continuous input space on a discrete output space, While minimising the loss of information. It works by encoding values from a multi dimensional vector space into a finite set of values from a discrete subspace of lower dimension. The lower space of vector requires less storage space, so the image can be easily compressed. Due to the density matching. property of vector quantizations the compressed data have errors that are inversely proportional to their density.[8]

In the process of vector quantization the image to be encoded is segment 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 requires because the reconstructed image highly depends on the codewords in this very codebook. The generated codebook is store into text file for vhdl file handing or data array in the vhdl code.

The algorithm is used for the design of optimal VQ is commonly referred to as Linde, Buzo and gray algorithm (LBG), It is based on the minimisation of the squared error distortion measure.

## II. LBG ALGORITHM

In 1980, Linde, Buzo and gray proposed this LBG algorithm especially designed for gray scale image compression and was proven to be an excellent tool for both speech and digital image compression.

LBG algorithm basically comprises of 3 most important steps of :-

1. Codebook generation.
2. Encoding procedure.
3. Decoding procedure.

in this codebook generation process, various image are divided into several K-dimension training vectors. The representative codebook is generated from these training vectors by the clustering techniques. Steps for codebook generation are as follows.:-

1) codebook design:-

Step 1:- Decompose image into non-overlapping sub-image blocks and termed as 'vectors'

Step 2:- A representation image subblock is selected to represent the entire set of image blocks.

Step 3:- Representative image vectors are c/a as 'codebook' and representative element image vector is called as 'code vector' .

2) Encoding:-

Step 1:- Take an image as input and select the highest level reduced image.

Step 2:- Decompose the obtained reduced image from step 1 into non-overlapping blocks of size nxn pixel and convert each block into vector.

Step 3:- Collect the vectors from steps and use them as an initial codebook for VQ process.

Step 4:- Improve the initial codebook into an improved one through several iterative process.

Step 5:- Store or transmit the codebook and the indextable as a compressed file of the input.

3) Decoding:-

Step 1:- Perform VQ decoding using the index table and the codebook to reconstruct the approximate image to original image.

Step 2:- The reconstructed image is replica of original image.

**Steps of the LBG algorithm :-**

- **Step 1**. First, you find the sample mean z1 (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 ≥ 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 $z1^{(1)}$ into two centroids, let $z_1^{(2)} = z_1^{(1)}$, $z_2^{(2)} = z_1^{(1)} + \varepsilon$, where ε is a small offset.
- **Step 5**. Use $\{z_1^{(1)}, z_2^{(1)}, \ldots, z_k^{(1)}\}$ 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.

## III. PROPOSED METHODOLOGY

Related computation is performed using MATLAB 7.11.0 We use MATLAB based image processing toolbox for generation of codebook. Since, the performance of the LBG algorithm is extremely dependent on the selection of the initial codebook. The initial codebook is choosen at random from the training data set. It is also seen that there might be the generation of poor quality codebooks. Generated codebook is stored into text file for VHDL file handling or data array in VHDL Code. Input is accepted only in the form of gray images which is further read in the MATLAB and stored into text file for VHDL file handling.

The complete available image is converted into 2 X2 non-overlapping image block size. Now, 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 on the basis of the computation of minimum squared Euclidean distance between the I/P 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 compressed form of input image [6].

In the reverse process, we perform reverse operation. Thus process is simple and straight forward where compressed image data is available.

The technique works on the gray scale image and the nature of code will be non-synthesis. For the VHDL file handling, input is provided in the form of image. The algorithm can be evaluated in terms of compression Ratio (CR), mean square error (MSE) and peak signal to noise ratio (PSNR).(12)

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. (12) Mathematically,

$$\text{Compression ratio (\%)} = \frac{\text{No of bits required before compression}}{\text{No of bits required after compression}}$$

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). (12) Mathematically,

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

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

The distortion in the decoded images is measured using peak signal to noise ratio, (12)

$$PSNR = 10\log_{10}\left( \frac{255^2}{\frac{1}{MXN} \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} (f(m,n) - g(m,n))^2} \right)$$

## IV. FLOW CHART OF THE PROJECT

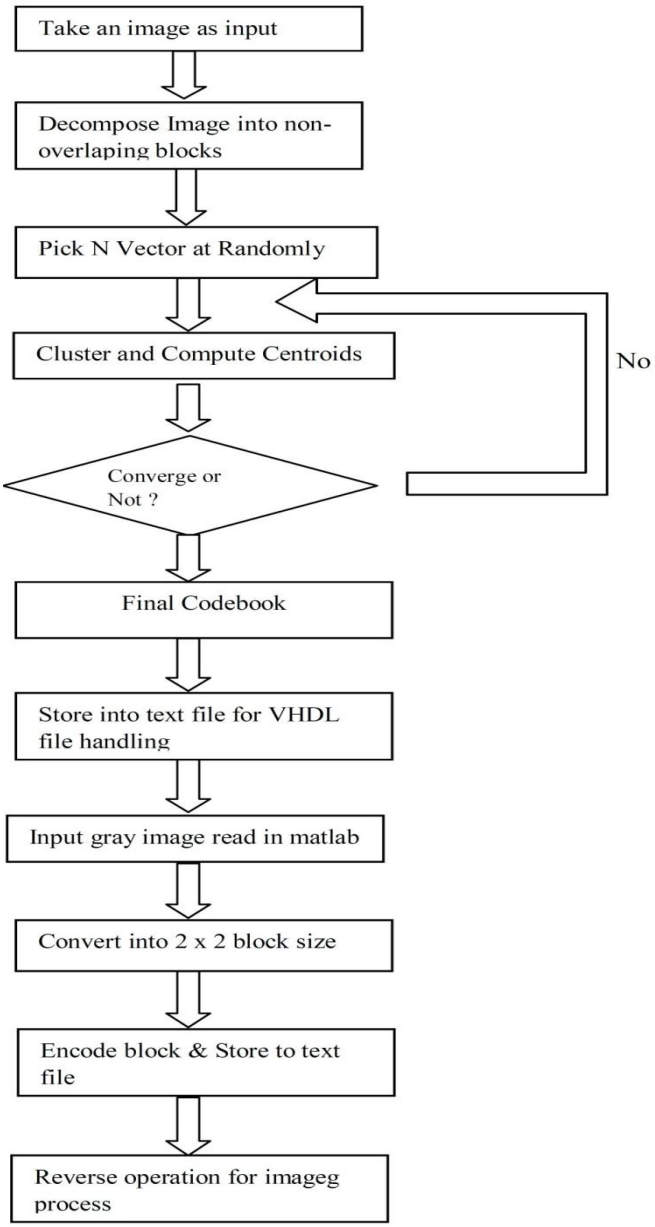



Figure: Flowchart of the project

**Explanation of Flow Chart**

- In order to compress image using VQ, initially image is decomposed into non-overlapping sub-image blocks, also termed as vectors, from all these vectors, a set of representative image vectors are selected to represent the entire set of image blocks. The set of representative image vectors is called a codebook and each representative image vectors is called as codeword. [6]
- Thus, the final codebooks are generated. The basic requirement of this algorithm is the codebook. We use MATLAB image processing toolbox for the related computation for the codebook generation.
- Generated codebook are then stored into text file for VHDL file handling
- Convert the whole image into 2 x2 Block size. 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 on the basic of 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. [6]

In Reverse operation at the receiver end, we translate the index back to its corresponding codeword. Thus, we obtain the reconstructed & compressed image.

## V. RESULT

Main aim of an image compression is to reduce image with the minimum number of bits of an acceptable image quality.

There are two major schemes for image compression.

1) Lossless image compression scheme
2) Lossy image compression scheme

Lossless image compression scheme is a reversible process where the exact reconstruction of the original image can be achieved, otherwise it is called as lossy image compression scheme. Lossy image compression scheme is a irreversible process but high compression can be easily achieved using lossy than lossless image compression scheme.

Scientific or legal consideration makes lossy compression unacceptable for high performance applications such as geophysics, telemetry, non-destructive, evaluation and medical imaging. Vector quantization is an established lossy compression scheme which is successfully used to compress signal such as speech, music and video.

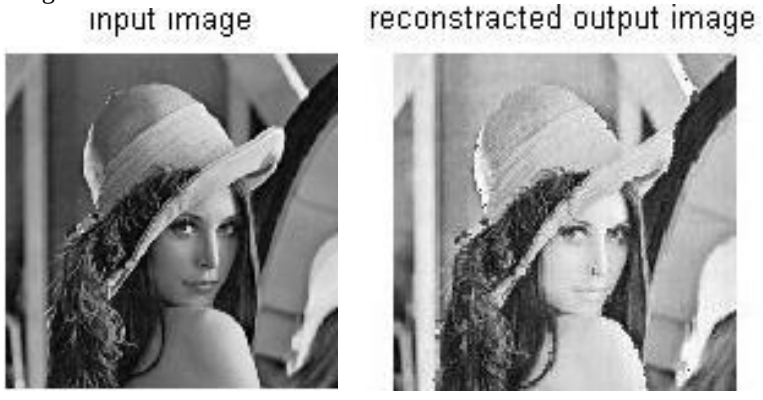
In the vector quantization, we group input samples into vector and using a vector quantizer, a lower bit rate and higher performance can be achieved.

Here, we use Linde, Buzo and gray algorithm for the designing of the VQ codebook. This algorithm is designed for the purpose of data compression. It minimum expected distortion in image compression. However, the performance of the standard LBG algorithm is highly dependent upon the choice of initial codebook. A training set of image is generally used for codebook generation. A single element of the codebook is called as "codevector" or "codeword". An initial codevector is set as the average of entire training sequence and are further split to provide to codevectors.

The algorithm is evaluated using Compression Ratio (CR) Peak Signal to Noise (PSNR) and Mean Square Error (MSE).Rate gives the number of bits per pixel which is used to encode an image. A typical grayscale image has a rate of 8 bits per pixel.(12)

$$\text{RATE(BPP)} = \frac{(8*\text{OUTPUT FILES SIZE[BYTES]})}{(\text{INPUT FILE SIZE [BYTES]})}$$

Here, we take an input image as 'Leena' and an output image called as "Reconstructed Image" is produced. Reconstructed image is a compressed image. The images are also supported with a Matlab based histogram. There are separate Histograms for both I/p image and reconstructed image. The project is an attempt to prepare error free, robust, scalable and compressed image.



**INPUT IMAGE**

**RECONSTRUCTED IMAGE**

Table: Compression measures

| Image | Codebook Size | Bit Used | PSNR | MSE | CR |
|---|---|---|---|---|---|
| Leena | 128 | 7 | 46.806 | 1.369 | 4:1 |
| FINGER PRINT | 128 | 7 | 45.15 | 14.97 | 4:1 |

## VI. CONCLUSION AND FUTURE SCOPE

### Conclusion

Data transfer of uncompressed image over digital networks require very high bandwidth. The state of art image compression techniques may exploit the dependencies between the sub-bands in a transformed image

The quality of the reconstructed image depends on the level of decomposition and size of the codebook. More the size of the codebook, better the quality. More compression is easily achievable if level of decomposition is increased.

Above proposed algorithm reduces the complexity of a transferred image without sacrificing the performance. The LBG algorithm is highly dependent upon the codebook generation.

It is an easy, rapid, efficient and simple algorithm which saves computation cost and time. Reduction can be also achieved by avoiding the computation of unnecessary codewords.

### Future Scope

Linde, Buzo & Gray Algorithm is an easy, rapid & efficient algorithm depending mainly upon the generation and quality codebooks. But LBG algorithm has the local optimal problem and the utility of each codeword in the codebook is low. The local optimal problem means that the codebook guarantees local minimum distortion but not global minimum distortion. It always converges to the nearest local minimum.

The other problem is that generation of codebook is a complex task. It needs more storage capacity for codebook. Convergence time is also very large.

The future enhancement primarily focuses on developing vector quantization algorithm that may ensure less storage space, less transfer time and less image viewing and loading time.

### REFERENCE

[1] Chin-Chen Chang and Yu-Chen-Hu, "Fast LBG codebook Training algorithm for vector quantization".0098-3063/98/IEEE.

[2] P.Franti, T.Kaukoranta, D.-F.Shen and K.-S.Chang, "Fast and memory efficient implementation of the exact PNN", IEEE Transactionsons Image Processing,9 (5), 773-777,May 2000.

[3] Peter Veprek ,A.B.Bradley, "An Improved Algorithm for vector quantizer design" 1070-9908 IEEE.

[4] Mukesh Mittal, Ruchika Lamba, "Image Compression Using Vector Quantization Algorithms : A Review", volume3, issue 6, june 2013 ISSN : 2277128 X

[5] 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.

[6] Arup Kumar Pal and Anupsar . "An Efficient codebook Initialization Approach for LBG algorithmic International Journal of Computer Science, Engineering and Application" Vol.1. No:4 August 2011.

[7] Manoj Kumar, Poonam Saini, "Image Compression with Efficient Codebook initialization using LBG Algorithm" ISSN: 2319-7900, IJACT.

[8] G.Boopathy, S. Arockiasamy, " Implementaion of Vector Quantization for Image Compression – A Survey" GJCST Computing classification 1.4.2, 1.2, 1.0, F.T.I. Page.22, Vol.10, Issue 3 (Ver 1.0) April 2010.

[9] Momotaz Begum, Nurun Nahar, Kaneez Fatimah and Md. Kamrul Hasan, " A New Initialization Technique for LBG Algorithm", 2nd International Conference on Electrical

[10] K.Masselos, T.Stouraitis, C.E. Goutis "NOVEL CODEBOOK GENERATION ALGORITHMS FOR VECTOR QUANTIZATION IMAGE COMPRESSION" IEEE International Conference Acoustic Speech and Signal processing. 1998, VLSI Design. Laboratory Department of Electrical and Computer Engineering. University of Patras Rio 26500, Greece.

[11] Qin Zhang, John M. Danskin, Neal E. Young "A Codebook Generation Algorithm for Document image Compression" 6211 Sudikoff laboratory, Department of Computer Science, Dartmouth College, Hanover, USA "Digital image processing" by S. Jayaraman, S. Esakkirajan, T. Veerakumar – Tata Mc Graw. Hill.

[12] "Digital image processing" by S. Jayaraman, S. Esakkirajan, T. Veerakumar – Tata Mc Graw. Hill