ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization) Vol.2, Special Issue 1, March 2014

Proceedings of International Conference On Global Innovations In Computing Technology (ICGICT'14)

Organized by

Department of CSE, JayShriram Group of Institutions, Tirupur, Tamilnadu, India on 6th & 7th March 2014

Optimization of Bit Rate in Medical Image Compression

Dr.J.Subash Chandra Bose[1], Mrs.Yamini.J[2], P.Pushparaj[3], P.Naveenkumar[4], Arunkumar.M[5], J.Vinothkumar[6]

Professor and Head, Department of CSE, Professional Group of Institutions, Palladam, India[1]

AP, Department of CSE, Professional Group of Institutions, Palladam, India[2]

Final year students, Department of CSE, Professional Group of Institutions, Palladam, India[3, 4, 5, 6]

ABSTRACT: The need for data compression has rapidly increased over the years. This is mainly due to the evolution of high speed networks and modern communication techniques. The main factors in multimedia and communication systems are the storage space and the time delay for transmission.
During the optimization process the modified Lloyds algorithm searches for empty cells i.e. the unused code vectors. Also it keeps track of the image vector producing maximum distortion corresponding to the code vector representing maximum number of image vectors are used to replace the unused code vectors thus removing the empty cells from the code book. As the iteration goes on the empty cells are completely removed.

The modified code book is free from empty cells it is more efficient than the LBG algorithm and produces a better picture quality. The performance measures can be compared using parameters such as Mean Square Error and Picture Signal to Noise Ratio.

KEYWORDS: Lloyd Algorithm, Huffman coding, Peak Signal to Noise Ratio. Vector Quantization, Empty cell problem

I. INTRODUCTION

1.1 Data Compression

The need far data compression has rapidly increased over the years. This is mainly due to the evolution of high speed networks and modern communication techniques. The main factors in multimedia and communication systems are the storage space and the time delay for transmission.

Image compression is one of the enabling technologies for each of these aspects of multimedia revolution. However compressing an image is significantly different than compressing raw binary data. Of course, general purpose compression programs can be used to compress images, but the result is less than optimal. This is because images have certain statistical properties which can be exploited by encoders specifically designed for them. Also, some of the finer details in the image can be sacrificed for the sake of saving a little more bandwidth or storage space. This also means that lossy compression techniques can be used in this area.

Here, the 'signal' is the original image, and the 'noise' is the error in reconstruction. So, if a compression scheme is found having a lower MSE (and a high PSNR), it is recognized as a better one.

1.1.1 Quantization

Quantization, involved in image processing, is a lossy compression technique achieved by compressing a range of values to a single quantum value. When the number of discrete symbols in a given stream is reduced, the stream becomes more compressible.

In many lossy compression applications we are required to represent each source output using one of a smaller number of code words. The number of possible distinct source output values is generally much larger than the number of codewords available to represent them. The set of inputs and outputs of the quantizers can be scalars or vectors. If they are scalars, we call the quantizers Scalar Quantizers. If they are vectors, we call the quantizers Vector Quantizers [2].

1.2 Scalar Quantization

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization) Vol.2, Special Issue 1, March 2014

Proceedings of International Conference On Global Innovations In Computing Technology (ICGICT'14)

Organized by

Department of CSE, JayShriram Group of Institutions, Tirupur, Tamilnadu, India on 6th & 7th March 2014

Scalar quantization is a lossy compression technique in which each input symbol is treated separately in producing the output.

A quantizer can be specified by its input partitions and output levels. If the input range is divided into levels of equal spacing, then the quantizer is termed as a Uniform Quantizer, and if not, it is termed as a Non-Uniform Quantizer. A uniform quantizer can be easily specified by its lower bound and the step size.

1.3Vector Quantization

Vector quantization is a quantization technique in which the basic idea is to code values from a multi- dimensional vector space into values from a discrete subspace of lower dimension.

The lower-space vector requires less storage space and the data is thus compressed. It is used in applications like image compression, voice compression and voice recognition. The vector quantization procedure is shown in Fig 1.1

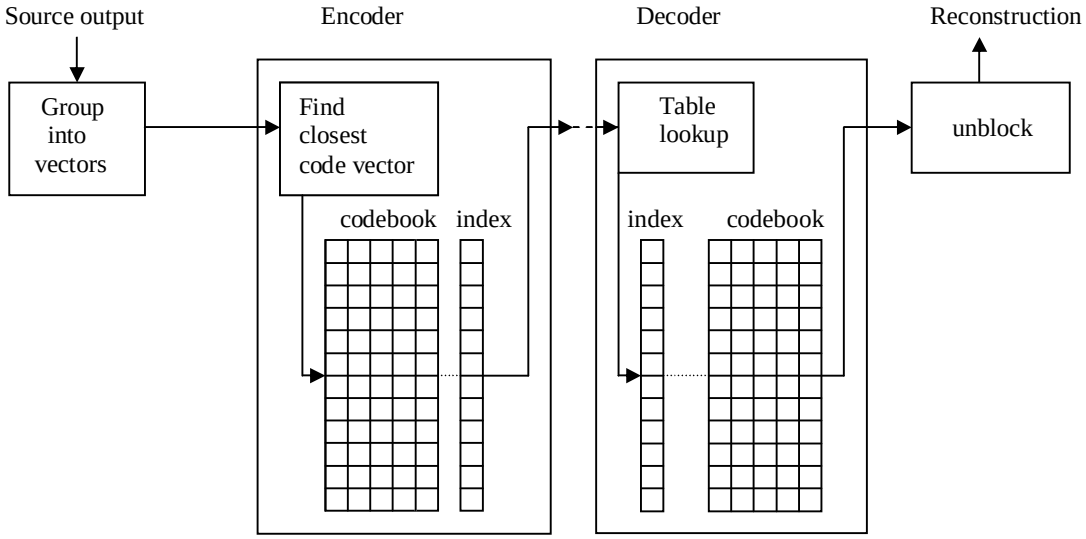


Fig 1.1 The vector quantization procedure

In vector quantization, we group the source output into blocks and vectors. This vector of source output forms the input to the vector quantizer called the codebook of the vector quantizer. The vectors in this codebook, known as code vectors, are selected to be representative of the vector we generate from the source output. Each code vector is assigned a binary index.

At the encoder, the input vector is compared to each code vector to find the code vector closest to the input vector.

In order to inform the decoder about which code vector was found to be the closest to the input vector, we transmit or store the binary index of the code vector. The decoder has exactly the same codebook and can retrieve the code vectors given in its binary index to reconstruct the image.

The codebook must contain vectors that represent well the images to be compressed. Several methods can be used for designing codebooks. They generally use a learning method on training sets issued from available images which are supposed to be representative of images to be compressed.

VQ has the particular advantage or disadvantage to have to exploit prior knowledge on images to be compressed. Since a codebook has to be generated before compression, one has to have a training set, i.e., several images that are representative of images to be compressed. Thus, VQ is not a "universal" approach (as e.g. PEG may be called), because it cannot work well for images of types too different from the training set ones. However, for images of the same type, it can work very well (better than general methods), because the codebook contains representative codewords.

Digital images in medical field verify the above mentioned condition. That is, every image of the same medical domain represents the same thing: the same part of the human body, the same organ, etc. Thus, a vey suitable codebook can be created and VQ can work very efficiently.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization) Vol.2, Special Issue 1, March 2014

Proceedings of International Conference On Global Innovations In Computing Technology (ICGICT'14)

Organized by

Department of CSE, JayShriram Group of Institutions, Tirupur, Tamilnadu, India on 6th & 7th March 2014

II. LLOYDS ALGORITHM

2.1 Introduction

The modified algorithm is used to overcome the Empty Cell Problem found in the LBG algorithm. The modified algorithm is very similar to the LBG Algorithm except for the optimization of the code vectors. As the additional code vectors are generated by adding offset to the existing code vectors the may not get mapped to any image vector and remain unused creating an Empty Cell.

During the optimization process the modified algorithm searches for empty cells i.e. the unused code vectors. Also the modified algorithm keeps track of the image vector producing maximum distortion corresponding to the code vector representing maximum number of image vectors. This maximum distortion image vectors are used to replace the unused code vectors thus removing the empty cells from the code book. As the iteration goes on the empty cells are completely removed and the final code book is free from empty cells.

As the modified code book is free from empty cells it is more efficient than the LBG algorithm and produces a better picture quality. The performance measures can be compared using parameters such as MSE (Mean Square Error) and PSNR (Picture Signal to Noise Ratio).

$$MSE = (1/MN) \sum_{y=1}^{M} \sum_{x=1}^{N} [I(x,y) - I'(x,y)]^2 \quad \text{.....(3.1)}$$

PSNR = 20 * log10 (255 / sqrt(MSE)) .....(3.2)

Where I(x,y) is the original image, I'(x,y) is the reconstructed version and M,N are the dimensions of the images.

2.2 Algorithm

The new image compression algorithm overcomes the Empty Cell Problem. It consists of two phases

a) Initialization of the code book.

b) Code book Optimization.

2.2.1 Code Book Initialization

The original GLA algorithm uses splitting technique to initialize the codebook. This technique basically doubles the size of the codebook in all iteration.

This procedure starts with one code vector, $C_1^{(0)}$ that is set to the average of all training vectors.

Step 1: In a general iteration there will be k code vectors in the code book. $C_i^{(0)}$ =1,2,….k.

Split each code vector into two code vectors $C_i^{(0)}$ and $C_i^{(0)}$ + r, where r is a fixed perturbation vector. Set k←2k.

Step 2: If there are enough code vectors, stop the splitting process. The current set of k code vectors can now serve as the initial set $C_i^{(0)}$ for the code book optimization phase. If more code vectors are needed, execute the optimistic algorithm on the current set of k entries, to converge them to a better set; then go to Step 1.

2.2.2 Code Book Optimization

Step 0: Select a threshold value 'Є', set k = 0 and $D^{(-1)} = \infty$. Start with initial code book with code vectors $C_i^{(k)}$ [where k is currently zero, but will be incremented in each iteration]. Training vectors are denoted as T_i.

Step 1: For each code vector $C_i^{(k)}$ find the set of all training vectors T_i, that satisfies equation (1),

$$d(T_i,C_i) < d(T_j,C_j) \; i \neq j$$

This set (or cell) is denoted as $P_i^{(k)}$. Repeat for all values of i. It may happen that some cells will be empty [Empty Cell Problem].

Step 2: Calculate the distortion $D_i^{(k)}$ between each code vector $C_i^{(k)}$ and the set of training vectors $P_i^{(k)}$ found for it in Step 1. Repeat for all i, then calculate the average D(k) of all the $D_i^{(k)}$. A distortion $D_i^{(k)}$ for a certain i is calculated by computing the distances $d(C_i^{(k)},T_m)$ for all training vectors T_m in the set $P_i^{(k)}$ then calculating the average distance.

Step 3: If $(D^{(k-1)}-D^{(k)}) / D^{(k)} \leq Є$, stop.

Otherwise, continue

Step 4: k = k+1, find new code vectors $C_i^{(k)}$ that are the average of training vectors in partition $P_i^{(k-1)}$ that was computed in Step 1.

Step 5: Check for Empty Cells and replace with image vectors producing maximum distortion. Go to Step1.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization) Vol.2, Special Issue 1, March 2014

Proceedings of International Conference On Global Innovations In Computing Technology (ICGICT'14)

Organized by

Department of CSE, JayShriram Group of Institutions, Tirupur, Tamilnadu, India on 6th & 7th March 2014

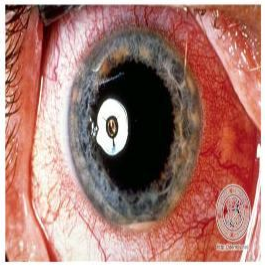
2.3 Implementation Using MATLAB 7.0

The new image compression algorithm has been simulated using MATLAB 7.0. First the optimized code book is generated from a set of four training images given in Table 2.1 and shown in Fig 2.1. Then the copy of the optimized code book is maintained at both the transmitter and the receiver.

Table 2.1 Training Images

S.No.	Training Images
1	REeye.jpg

REeye.jpg



2.3.1 Transmitter

At the transmitter the optimized code book is used to compute the index for any image that comes into it. The image is first divided into image vectors and then each image vector is compared with all the code vectors in the code book. The one which produces minimum distortion with the image vector is assigned to the index. The index matrix is Huffman encoded and transmitted to the receiver.

2.3.2 Receiver

At the receiver Huffman decoding is done to get back the index matrix. Huffman encoding is lossless in nature and we get the same index as computed by the transmitter. The original image is reconstructed by taking the appropriate code vectors from the code book as indicated by the index.

2.4 Region Of Interest

In medicine, images are used for a well-defined task: the analysis of different parts of the human body. Moreover, in certain images, different regions may have various levels of diagnostic importance: pathologies are present, and visible, in specific regions of the image, corresponding to well defined organs or parts of organ.

For example, in an X-ray hand image the expert is mainly interested in bones. These bones occupy finally a very restricted part of the image; therefore the parts of the image containing no bones, i.e. a large image zone, may be compressed with a low bit rate because a poorer reconstruction quality is acceptable.

In this method, a separate codebook is generated for each region (which may be an organ, a specific 'object'). The properties of these codebooks, i.e. the codebook size and the block size, are chosen according to the medical importance of the given region. For a region which is important for diagnosis, a large codebook containing small code words is created. On the other hand, for less important regions, block sizes are smaller and/or codebooks contain less code vectors.

First of all, image analysis and segmentation has to be performed in order to determine regions, to locate 'objects'. Since a strong prior knowledge is exploitable, specified and efficient segmentation techniques can be developed for a given image type.

Image analysis results in a model, i.e. the global structure of the image. This description has to be transmitted to the decoder, and is used to guide the compression process.

These data do not take in general a large amount of memory. According to the segmentation information resulting from image analysis, encoder and decoder always use the appropriate codebook. Thus, the image is compressed with a high quality wherever it is required, and with a low bit rate whenever a higher image distortion is allowed [1].

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization) Vol.2, Special Issue 1, March 2014

Proceedings of International Conference On Global Innovations In Computing Technology (ICGICT'14)

Organized by

Department of CSE, JayShriram Group of Institutions, Tirupur, Tamilnadu, India on 6th & 7th March 2014

2.5 Huffman Coding

Huffman coding is a method of lossless data compression, and a form of entropy encoding. The basic idea is to map an alphabet to a representation for that alphabet, composed of strings of variable size, so that symbols that have a higher probability of occurring have a smaller representation than those that occur less often.

Huffman's algorithm constructs an extended binary tree of minimum weighted path length from a list of weights. For this problem, our list of weights consists of the probabilities of symbol occurrence. From this tree (which we will call a Huffman tree for convenience), the mapping to our variable-sized representations can be defined.

The Huffman coding method leads to the lowest possible value of L bar for a given M, resulting in a maximum η. Hence it is also known as minimum redundancy code, or optimum code. The procedure is as follows

- N messages are arranged in an order of increasing probability.
- The probabilities of [N-K(M-1)] least likely messages are combined, where k is the highest integer that gives positive value to the bracket, and the resulting [K(M-1)+1] probabilities are rearranged in a non-increasing manner. This is called reduction.
- The reduction procedure is repeated as often as necessary, by taking m terms every time, until there remain M ordered probabilities. It may be noted by combining [N-K(M-1)],and not M, terms in first reduction it is ensured that there will be exactly M terms in the last reduction.
- Encoding begins with the last reduction, which consists of exactly M ordered probabilities. The first element of encoding alphabet is assigned as the first digit in the code words for all source messages associated with the first probability of the last reduction. Similarly the second element of the encoding alphabet is assigned as the second digit in the code words for all source messages associated with the second probability of last reduction and so on.

The same procedure is repeated for the second from last reduction, to the first reduction, in that order. Illustration is shown below. For a simple example, we will take a short phrase and derive our probabilities from a frequency count of letters within that phrase. The resulting encoding should be good for compressing this phrase, but of course will be inappropriate for other phrases with a different letter distribution.

We will use the phrase "math for the people by the people". The frequency counts of characters in this phrase are as.

follows (let ␣ denote the spaces). We will simply let the frequency counts be the weights. If we pair each symbol with its weight, and pass this list of weights to Huffman's algorithm, we will get something like the following tree (edge labels have been added). From the tree shown in Fig 2.2, we obtain the following mapping shown in Table 2.3ig 2.2 Huffman coding tree

Table 2.2 Frequency count of characters

Letter	Count
␣	6
e	6
p	4
h	3
o	3
t	3
l	2
a	1
b	1
f	1
m	1
r	1
y	1
Total	33

Table 2.3 Huffman code

Letter	Count	Huffman code
␣	6	111
E	6	01
P	4	101
H	3	1100
O	3	1101
T	3	001
L	2	0001
B	1	00001
F	1	10000
M	1	10001
R	1	10010
Total	31	-

If we were to use a fixed-sized encoding, our original string would have to be 132 bits in length. This is because there are 13 symbols, requiring 4 bits of representation, and the length of our string is 33.

The weighted path length of this Huffman tree is 113. Since these weights came directly from the frequency count of our string, the number of bits required to represent our string using this encoding is the same as the weight of 113. Thus

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization) Vol.2, Special Issue 1, March 2014

Proceedings of International Conference On Global Innovations In Computing Technology (ICGICT'14)

Organized by

Department of CSE, JayShriram Group of Institutions, Tirupur, Tamilnadu, India on 6th & 7th March 2014

the Huffman encoded string is 85% the length of the fixed-sized encoding. Arithmetic encoding can in most cases obtain even greater compression, although it is not quite as simple to implement.

2.6 Comparison metrices:

The following are the list of metrics used to analyze the performance of the new image compression algorithm in comparison with the existing LBG algorithm.

Table 2.4 Performance Metrics

S.No.	Performance Metrics	Formula for computing
1	Mean Square Error (MSE)	$(1/MN) \sum_{y=1}^{M} \sum_{x=1}^{N} [I(x, y) - I'(x, y)]^{\wedge}2$
2	Picture Signal to Noise Ratio (PSNR)	$10 \log (255^2/MSE)$
3	Bit Rate	Bits(Compressed Image) / Number of Pixels
4	Compression Ratio	Bytes(Compressed Image) / Bytes(Original Image)
5	Compression Rate	Bytes(Ori Img)-Bytes(Comp.Img.) / Bytes(Original Image)

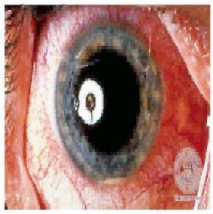
III. EXPERIMENTAL RESULTS

3.1 Comparison of Bit Rate:

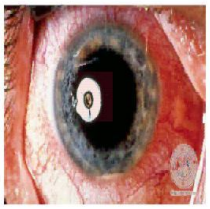
Comparison of Bit Rate	Training Images	LBG Algorithm	Proposed Algorithm
Code Book Size: 512 Code vector size: 4 x 4	REeye	1.5090	1.6582
Code Book size : 512 Code vector size: 8 x 8	REeye	0.4458	0.4790
Code Book size : 256 Code vector size: 4 x 4	REeye	1.3484	1.4326
Code Book size : 256 Code vector size: 8 x 8	REeye	0.3640	0.3904

Table 3.1 (b) Bit Rate for 512 Code book 4 x 4 code vectors

3.2 Comparison of Bit rate: (512 code vector)



LBG – 4 x 4 cv



Proposed – 4 x 4 cv

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization) Vol.2, Special Issue 1, March 2014

Proceedings of International Conference On Global Innovations In Computing Technology (ICGICT'14)

Organized by

Department of CSE, JayShriram Group of Institutions, Tirupur, Tamilnadu, India on 6th & 7th March 2014

Original Image 256x256	 LBG – 8 x 8 cv	 Proposed – 8 x 8 cv

Fig 3.2 REeye.jpg using 512 code vectors

3.3 Comparison of Bit rate: (256 code vector)

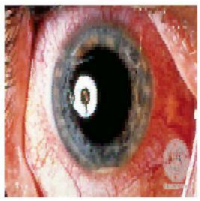
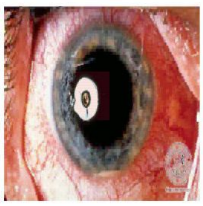
 Original Image 256x256	 LBG – 4 x 4 cv	 Proposed – 4 x 4 cv
	 LBG – 8 x 8 cv	 Proposed – 8 x 8 cv

Fig 3.3 REeye.jpg using 256 code vectors

IV. CONCLUSION

We have presented an image compression method, which deeply exploits the advantages of VQ in medical application environment. Based on prior knowledge and using Regions of Interest, our approach compresses relevant regions with a very good reconstruction quality. Moreover, a rather high overall compression rate is obtained due to the strong compression in less important image zones. Since separate VQ codebooks are created to compress different objects, every codebook is well adapted to the given region.

Simulation results provided by compressing the studied images validate the interest of the proposed approach. A good performance is obtained, and the quality is preserved on the most important part, i.e. on the diagnostically important region. In future, the region of interest which is lossily compressed in the proposed algorithm can be losslessly compressed. This will result in a better image quality but the compression rate will be low.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization) Vol.2, Special Issue 1, March 2014

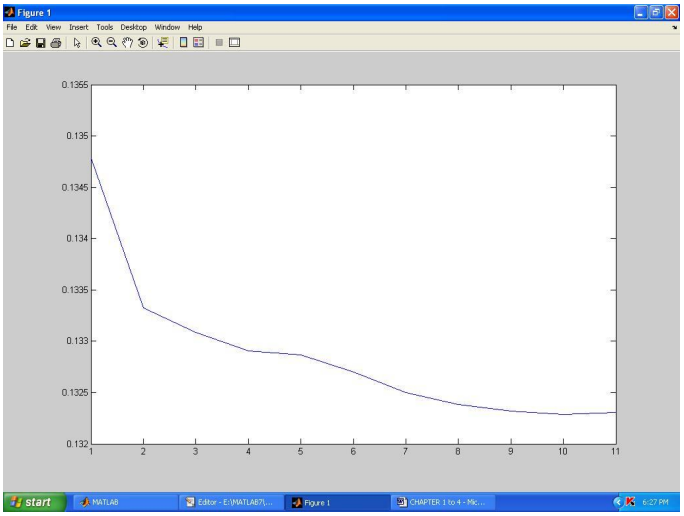
Proceedings of International Conference On Global Innovations In Computing Technology (ICGICT'14)

Organized by

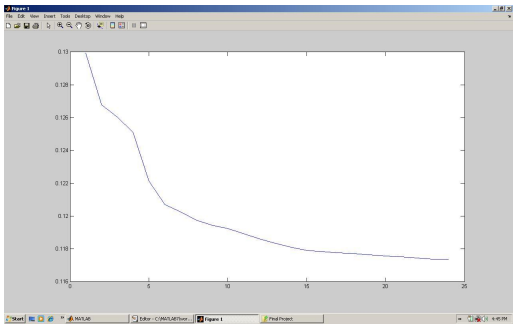
Department of CSE, JayShriram Group of Institutions, Tirupur, Tamilnadu, India on 6th & 7th March 2014

V. SCREEN SHOTS

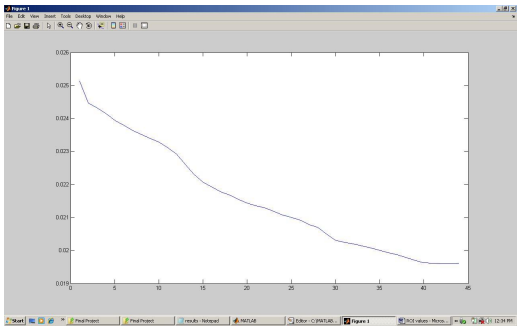
B.1 OPTIMIZATION CURVE FOR 256 CODE BOOK 4 X 4 CODE VECTORS USING LBG ALGORITHM



B.2 OPTIMIZATION CURVE FOR 256 CODE BOOK 4 X 4 CODE VECTORS USING PROPOSED ALGORITHM



B.3 OPTIMIZATION CURVE FOR ROI 512 CODE BOOK 2X2 CODE VECTORS USING PROPOSED ALGORITHM



REFERENCES

[1] Cazugue1.G, Czih.A, Solaiman.B, Roux.C, "Medical image compression and analysis using Vector Quantization, the Self-organizing Map, and the quad tree decomposition", Conference on Information Technology Applications in Biomedicine, Washington, USA, May 1998.

[2] Rafael C. Gonzale, Richard E. Woods (2003), "Digital Image Processing", Second Edition, Pearson Education, July 2003.

[3] Vaisey.J, Gersho.A, "Image Compression with variable Block Size Segmentation", IEEE Transactions on Signal Processing, Vol1.40, No 8, August 2004.

[4] Vlaciu.A, Lungu.S, Crisan.N, and Persa.S, "New compression techniques for storage and transmission of 2-D and 3-D medical images" In Advanced Image And Video Communications and Storage Technologies, volume 2451, pages 370-7.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization) Vol.2, Special Issue 1, March 2014

Proceedings of International Conference On Global Innovations In Computing Technology (ICGICT'14)

Organized by

Department of CSE, JayShriram Group of Institutions, Tirupur, Tamilnadu, India on 6th & 7th March 2014

[5] [Online] http://www.mic.ki.se/MEDIMAGES.html

BIOGRAPHY

Dr. J. SUBASH CHANDRA BOSE.
DEEE.,BE(EEE).,ME(ICE).,PhD(ICE).,

He is having more than 11 years of experience in both teaching and industry. Presently working as Professor and Head, Department of Computer Science and Engineering in Professional Group of Institutions, Coimbatore. His research area is Medical Image processing especially detection of cancer. He published several journals in both National and International journals.