ISSN (Print) : 2319-5940
ISSN (Online) : 2278-1021

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 3, Issue 1, January 2014

Iteration Free Fractal Color Image Compression Using Vector Quantization

A R Nadira Banu Kamal[1], P.Priyanga[2]

Head, Prof. Dept of Computer Science, TBAK College for Women, Kilakarai, Tamil Nadu, India[1]

Project Fellow, Dept of Computer Science, TBAK College for Women, Kilakarai, Tamil Nadu, India[2]

Abstract: The storage requirements for images can be excessive, if true color and a high-perceived image quality are desired. An RGB image may be viewed as a stack of three gray-scale images that when fed into the red, green and blue inputs of a color monitor, produce a color image on the screen. By convention the three images forming an RGB color image are referred to as the red, green and blue component images. The abnormal size of many images leads to long, costly, transmission times. Cost is not the only reason for speeding up transmission. The emergence of the World Wide Web has resulted in a huge increase in the exchange of image via the Internet. Image compression is method for proficiently coding digital image by minimizing the number of bits need for denoting image. It is main goal is to reduce the storage space and cut down the transmission cost and maintain good quality. The methodology used in this present work is to reduce the coding process time, intensive computation tasks and also memory requirements. The redundancies in the domain pool are reduced by the Linde Buzo Gray (LBG) Algorithm. The domain pools that are more efficient than those in the conventional fractal coding and the proposed iteration-free schemes achieve excellent performances.

Keywords: Domain Pool, Fractal Image Compression, Iteration Free Fractal Code, LBG Algorithm

I. INTRODUCTION

The storage requirements for images can be excessive, if true color and a high-perceived image quality are desired. Storage problems are acute in remote sensing applications, the scenes imaged by earth – orbiting satellites have widths and heights of several thousand pixels and there may be several bands representing the different wavelengths in which images are acquired. The raw data for a single scene therefore requires several hundred megabytes of storage space. Certain types of redundancy occur in an image. Image compression is achieved by removing these redundancies. If the image after decompression is exactly the same as the original image then compression is said to be lossless otherwise it is said to be lossy. The proposed method deals with lossy compression. Different techniques are employed to achieve lossy compression. Fractal compression is one of the methods and this method is used in this research work to achieve improved image quality, compression ratio and reduction in coding time.The fractal image compression problem puts forward three major requirements: speeding up the compression algorithm, improving image quality and increasing compression ratio [1]. There are many modified versions proposed to improve the fractal coding techniques [2]. Most of the studies focus on the type of the image partition [3], reducing of the complexity of the encoding process [4], speeding up the process [5, 6], LBG coding technique [7].

The domain pool used in fractal compression is often referred to as a virtual codebook, in comparison with the codebook of Vector Quantization. Increased fidelity is obtained by allowing search over a larger set of domains but this increases the time consumption and the number of bits required to specify the selected domain. The different types of codebook used are global codebook, local codebook and synthetic codebook. In the proposed method synthetic codebook is used with the domain blocks placed at the intervals equal to that of the range block width. These domain blocks are not as effective in matching the range blocks as those derived in the usual fashion but the decoding does not require iteration hence there is a reduction in time and the coding error is determined immediately at the encoder. The size of the VQ blocks is usually kept very small which in turn results in a high statistical correlation between adjacent blocks. In this research work the size of the vector is limited to 64 ie., 8x8 blocks. They usually have structural characteristics like surface smoothness and edges. The searching process of the reproduction vector for an input vector using full search requires intensive computations.

A. Vector Quantization for Image Compression

The proposed methodology using the VQ technique reduces the coding process time and intensive computation tasks by pruning the domain block for each range block. The

ISSN (Print) : 2319-5940
ISSN (Online) : 2278-1021

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 3, Issue 1, January 2014

redundancies in the domain pool are first reduced by the Linde Buzo Gray (LBG) Algorithm. Further redundancy in the domain block for each range block was achieved using the vector features such as mean value, edge strength, and texture strength. A pruning condition for terminating the searching process to find the best domain block from the domain pool has been used in this proposed research work. The main goal is to accelerate image compression without significant loss of image quality and with an acceptable compression rate.

These techniques in the fractal image coding have produced a good image quality at a good compression ratio with reduction in computation complexity and encoding time.

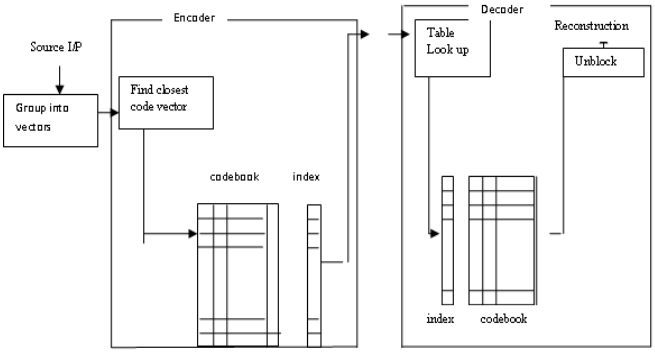


Fig 1.The Vector Quantization Procedure

B. LBG Algorithm

This algorithm also known as *Generalized Lloyd's Algorithm* is the basis of most vector quantization designs

Step1: Start with an initial set of reconstruction value $\{y_i^{(0)}\}_{i=1}^{M}$ and set of training vectors as $\{X_n\}_{n=1}^{N}$. Initialize k=0 and, $D^{(0)}$ to some high value. Select the threshold ϵ.

Step2: The quantization region $\{V_i^{(t)}\}_{i=1}^{M}$ are given by $V_i^{(t)} = \{x_n : d(x_n, y_i) < d(x_n, y_j) \quad \forall j \neq i\} \quad i = 1,2,\ldots M$. None of the quantization region should be empty.

Step3: Calculate the average distortion $D^{(k)}$ between the training vectors and the representative reconstruction value.

Step4: If $((D^{(k)} - D^{(k-1)}) / D^{(k)}) < \epsilon$ stop, otherwise continue.

Step 5: Increment the value of k by 1. Find new reconstruction value $\{y_i^{(k)}\}_{i=1}^{M}$ which is the average value of the element of each of the quantization regions $V_i^{(k-1)}$. Go to step 2.

The LBG Algorithm guarantees that the distortion from one iteration to the next will not increase. However there is no guarantee that the procedures will converge to the optimal solution. The solution to which the algorithm converges is heavily dependent on the initial condition.

Linde, Buzo and Gray described a technique called the splitting technique for initializing the design algorithm. In this technique of designing a vector quantizer, a single output vector is first designed, in other words a codebook of size one or a one-level vector quantizer. With a one-element codebook, the quantization region is the entire input space, and the output vector is the average value of the entire training set. From this output vector the initial codebook for a two-level vector quantizer can be obtained by including the output point for the one – level quantizer and a second output vector obtained by adding a fixed perturbation vector ε. Then the LBG algorithm is used to obtain the two-level vector quantizer. Once the algorithm has converged the two codebook vectors are used to obtain the initial codebook of a four-level vector quantizer. This initial four-level codebook consists of the two codebook vector from the final codebook of the two-level vector quantizer and other two vectors obtained by adding ε to the two codebook vectors.

II. ARCHITECTURE AND ALGORITHM OF THE PROPOSED TECHNIQUE

The iteration-free fractal coding does not require iteration at the decoding stage which contributes to reduction in time. For this purpose synthetic codebook is used. This codebook is not transmitted by the encoder either online or offline. In the proposed method, the synthetic codebook is constructed using the mean image, whose pixel values are the block means of all the range blocks. We use the mean information of the range blocks that are hidden in the modified contractive affine transformation of the fractal codes to obtain the same domain blocks in both the encoder and decoder without using an off-line transmission of the codebook. The architecture of the proposed method is described in Fig. 2.

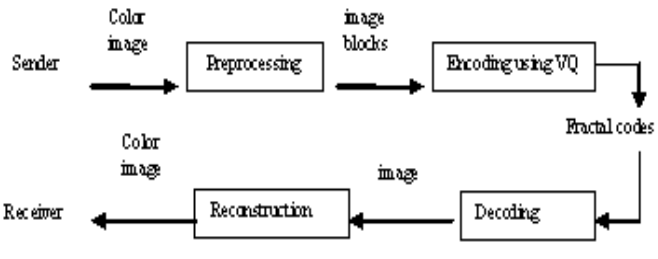


Fig. 2 Architecture of the Proposed Iteration-Free Fractal Image Coding Method

ISSN (Print) : 2319-5940
ISSN (Online) : 2278-1021

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 3, Issue 1, January 2014

The sender sends the (R, G, B) color image for compression. In the preprocessing stage, the input MxNx3 (512x512x3) image under coding is divided into non-overlapping square blocks of BxBx3 pixels called the range blocks. Then the mean and variance of each range blocks are determined. After the mean of all the range blocks are obtained, a mean image of size M/B x N/B with each pixel corresponding to the block mean is generated. The mean image must be larger than the size of the range block i.e. M/B x N/B > B x B. The maximum size of B is limited to 8 in order to produce a good quality of the decoded image. The higher the resolution of the input image (MxNx3) more blocks can be generated for the domain pool which helps to find a good mapping between the domain and range blocks. The initial domain pool with blocks of the same size as the range is generated using the mean image. In the encoder if the variance of the range block is smaller than the threshold value *E*, the range block is coded by the mean, or else the range block will be coded by the contractive affine transformation. The aim of the proposed scheme is to find the domain block for each image range block and the transformation parameters that minimize the distortion between the image block and the transformed domain block in a minimized time. This process of finding the best domain block makes use of the techniques like VQ.

In this proposed method, the LBG algorithm and vector features are applied to reduce the redundancies in the generated domain blocks of the domain pool. The numbers of calculations to determine the best domain block is reduced by extracting the features of the range block like mean, edge strength and texture strength and comparing it with the domain pool and eliminate redundant domain blocks. The transformations are applied only to the remaining domain blocks and the transformation parameters that minimize the distortion between the image block and the transformed domain block is coded. The architecture of the proposed encoder using VQ method is described in Fig. 3.

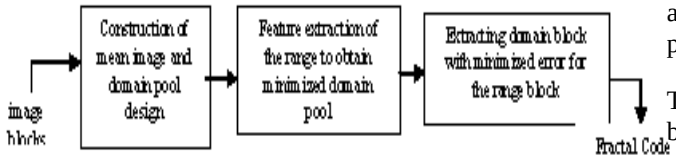


Fig. 3.Proposed Encoder Using VQ

In the decoder, shown in Fig. 4 the mean information of each range block is extracted from the fractal codes. Using this information the mean image is constructed. This mean image is partitioned into blocks of the same size as the input image. This forms the domain pool for VQ the domain pool is constructed from the mean image blocks (same size as that of the input) using LBG algorithm. The decompressed image is constructed block by block by applying the transformation parameters to the selected domain block from the domain pool as per the code.

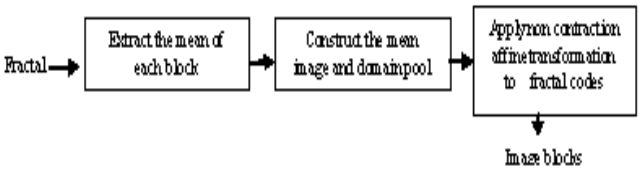


Fig.4 Proposed Decoder

III. ALGORITHM OF THE ENCODER FOR THE PROPOSED ITERATION-FREE FRACTAL IMAGE CODING USING VQ

A. Proposed Encoder

In the encoder the input image is partitioned into blocks. The mean and variance of each block is calculated. The initial domain pool is generated using the mean image. In the encoder if the variance (3.1) of the range block

$$V\{R\} = \frac{1}{B^2} \sum_{0 \le i,j < B} (r_{i,j} - \mu_R)^2 \qquad (3.1)$$

(where $r_{i,j}$ denotes the (i, j)th pixel in the range block of size BxB) is smaller than the threshold value *E*, then the range block is coded by the mean or else the range block will be coded by the contractive affine transformation. Given the mean of each range block and the set of block transformations, the proposed scheme finds for each image block the domain block and the transformation parameters that minimize the distortion between the image block and the transformed domain block. For N domain blocks (vectors of size k), N distortion computations are needed to determine the best match of an input range block. For a large number of domain blocks, the determination process is very time consuming. To keep the distortion almost the same as achieved by full search and to speed up the encoding process, partial domain block searches are used.

The new contractive affine transformation can be expressed by

$$\hat{R} = i\{\alpha.D + \mu_R - \alpha.\mu_D\} = i\{\alpha.(D - \mu_D) + \mu_R\}$$

$$= i\{\ \alpha\ .\ (W_i - \text{mean}\ (W_i)) + \text{mean}(R)\} \qquad (3.2)$$

The transformations applied to the domain pool are contrast scaling (α) and isometries (*i*). The size of the domain blocks D is the same as that of the range block R and thus the contraction procedure in fractal coding schemes is eliminated. Therefore a new contractive affine

ISSN (Print) : 2319-5940
ISSN (Online) : 2278-1021

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 3, Issue 1, January 2014

transformation between the range block and the domain blocks in the domain pool is calculated using Eq. 3.2. The parameters used in the new contractive affine transformation are specified as follows.

The main aspect of fractal-based image coding is to find the position of a suitable domain block P_D and a transformation for a rough type range block. The luminance shift is replaced by the mean μ_R. The contrast scaling α is determined by testing all the values in the following set {n/4, n=1, 2, 3, 4} to find the best one that minimizes the distortion. If the domain blocks are represented by $D_{i,j}$ the eight canonical isometries transformation (i_0 - i_7) of the domain blocks are obtained as follows.

a) Identity:

$$i_0(D_{i,j}) = D_{i,j} \quad (3.3)$$

b) Orthogonal reflection about mid-vertical axis (j = (B – 1)/2) of block:

$$i_1(D_{i,j}) = D_{i,B-1-j} \quad (3.4)$$

c) Orthogonal reflection about mid-horizontal axis (i = (B-1)/2) of block:

$$i_2(D_{i,j}) = D_{B-1-i,j} \quad (3.5)$$

d) Orthogonal reflection about first diagonal (i = j) of block:

$$i_3(D_{i,j}) = D_{j,i} \quad (3.6)$$

e) Orthogonal reflection about second diagonal (i + j = B - 1) of block:

$$i_4(D_{i,j}) = D_{B-1-j,B-1-i} \quad (3.7)$$

f) Rotation around center of block, through +90°:

$$i_5(D_{i,j}) = D_{j,B-1-i} \quad (3.8)$$

g) Rotation around center of block, through +180°:

$$i_6(D_{i,j}) = D_{B-1-i,B-1-j} \quad (3.9)$$

h) Rotation around center of block, through -90°:

$$i_7(D_{i,j}) = D_{B-1-j,i} \quad (3.10)$$

An advantage of coding using isometry transformation is that it can be decoded to any size either enlarged or minimized depending on the requirements.

The encoding procedure can be summarized in the following steps. Each one is calculated separately for R, G, and B and combined for further result.

Step1: The mean μ_R and variance V_R of each range block R(i,j) is determined. A mean image of size M/B x N/B with each pixel corresponding to the block mean is generated. The mean image must be larger than the size of the range block i.e. M/B x N/B > BxB.

Step2: The mean image is divided into blocks of the same size as the range block (B x B pixels) to form the domain pool.

Step3: If the variance of the range block is smaller than the threshold value *E*, then the range block (smooth block) is coded by the mean, otherwise, the range block (rough block)

will be coded by the fractal code $f(i, \alpha, \mu_R, P_D)$ using VQ / GA / SA techniques. Here *i* represent the isometry transformations, α the contrast scaling, μ_R the mean value of the range block and P_D the domain block number in the domain pool.

$$\hat{R}(i,j) = \mu_R \quad \text{if } V_R \le E$$
$$= f(i, \alpha, \mu_R, P_D) \quad \text{if } V_R > E \quad (3.11)$$

Thus the input from the sender is the image and the output is the fractal codes.

The redundant domain blocks are initially eliminated using LBG algorithm as in the existing iteration-free fractal coding using VQ. The enhancement proposed in this method using VQ is to further reduce the domain blocks that are redundant by examining the vector features. The following method helps in identifying the domain blocks that are eliminated in the search.

Let v_1,v_2 and v_3 be three orthogonal vectors, where

$$v_1 = ¼ [1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1] \quad (3.12a)$$

$$v_2 = ¼ [1,1,1,1,1,1,1,1,-1,-1,-1,-1,-1,-1,-1,-1] \quad (3.12b)$$

$$v_3 = ¼ [1,1,-1,-1, 1,1,-1,-1,1,1,-1,-1,1,1,-1,-1] \quad (3.12c)$$

for the codebook of size k=16. The axis in the direction of v_i (i=1, 2, 3) is denoted as the ith axis. Let x_i be the projection value of an input block (vector) X on the i^{th} axis. That is, x_i is the inner product of X and v_i which is calculated, as follows:

$$x_i = \langle X, v_i \rangle \quad (3.12d)$$

ISSN (Print) : 2319-5940
ISSN (Online) : 2278-1021

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 3, Issue 1, January 2014

Similarly, denote c_{ki} as the projection value of a domain block W_k on the i^{th} axis. To speed the searching process, all domain block are sorted in ascending order of their projections on the first axis. Here x_1 is four times the mean value of X; x_2 and x_3 are the edge gradients in the vertical and horizontal directions, respectively, of X; and $[(x_2)^2 + (x_3)^2]$ represent the edge strength of X. Similar meanings are applied to c_{ki} (i=1,2,3). Let r be the distance between an input block (vector) X and a domain block W_j. If domain block W_j cannot satisfy the following condition, it will be rejected directly in the process of finding the closest domain block of X

$$\left| c_{ji} - x_i \right| < r, \text{ i=1, 2, 3} \qquad (3.13a)$$

where x_i and c_{ji} are the projection values of X and W_j, respectively, on the i^{th} axis. As shown in condition (3.13a), a smaller value of r will give a better performance of rejecting unlikely domain block. If the domain block W_i is the closest domain block of X, then their projection values on the first axis may be very close. As stated before, the projection value on the first axis of a vector is four times the mean value of the vector. Therefore, the domain blocks W_i, whose mean value is close to the mean value of X, is chosen as the initial domain blocks for that range block. An additional condition to reduce the distortion computations is also used. To reject irrelevant domain block, the following condition with condition (3.13a) is used to reject unlikely domain block in the process of finding the closest domain block of an input range block. Let c_j be the projection of the domain block W_j on the space spanned by v_1, v_2 and v_3, where

$$c_j = c_{j1}v_1 + c_{j2}v_2 + c_{j3}v_3 = \sum_{i=1}^{3} \langle W_j, v_i \rangle v_i$$

(3.13b)

Similarly, denote x as the projection of the input range vector X on the space spanned by v_1, v_2 and v_3, where $x = x_1v_1 + x_2v_2 + x_3v_3 = \sum_{i=1}^{3} \langle X_j, v_i \rangle v_i$. Let $s_x = X - x$ and $s_{cj} = W_j - c_j$. From the definitions of c_j, x, s_{cj} and s_x, $c_j \perp s_{cj}, c_j \perp s_x, x \perp s_{cj}$ and $x \perp s_x$. A candidate domain block W_j should satisfy the following condition:

$$\left[(c_{j1} - x_1)^2 + (c_{j2} - x_2)^2 + (c_{j3} - x_3)^2\right] + \left(|s_{cj}| - |s_x|\right)^2 < r^2 \qquad (3.14)$$

That is, if the domain block W_j cannot satisfy condition (3.14), it will be discarded directly in the process of finding the closest domain block of X. Condition (3.14) activates only when the domain block cannot be rejected by using condition (3.13a). The texture vector (block) has a small value of $[(x_2)^2 + (x_3)^2]$ and a large value of $(|s_x|)^2$, which is called the texture strength of X; an edge block X posseses a large value of $[(x_2)^2 + (x_3)^2]$ and a small value of $(|s_x|)^2$ and a smooth block X gives a small value of $[(x_2)^2 + (x_3)^2]$ and $(|s_x|)^2$. The x_1 is four times the mean value of X. The same characteristics are also applied to all domain blocks. A smooth domain block mainly uses its projection value on the first axis to distinguish itself from other smooth domain block; an edge domain block distinguishes itself from other edge domain block using all three projection values; a texture domain block uses the texture strength and the projection value on the first axis to distinguish itself from other texture domain block. That is, (3.14) uses three features namely mean value, edge strength, and texture strength of a vector to reject unlikely domain block. Therefore, (3.14) has a good performance of rejecting unlikely domain block for an input range block if a good initial domain block is found. Another condition for terminating the searching process, if the distance r between a domain block W_i and input range block X is smaller than half the distance between W_i and any other domain block, then the domain block W_i must be the best match of the training vector X. Thus, the searching process may be stopped and W_i may be chosen as the closest domain block when it satisfies (3.15)

$$d(X, W_i) \leq 0.5 \min (d(W_j, W_i), j=1,2,\ldots,i-1,i+1,\ldots N) \qquad (3.15)$$

Let $d_{ni} = 0.5 \min (d(W_i, W_j), j=1,2,\ldots,i-1,i+1,\ldots N)$, where d_{ni} is half the minimum distance between W_i and all other domain block. Thus using the above method the best domain block for each of the range block can be determined quickly. The new contractive affine transformation can be calculated using Eq. 3.2

The distortion between the original range block (R) and the coded range block ($\hat{R}$) is represented by the mean-squared-error (MSE) measurement defined as:

$$MSE(R, \hat{R}) = \frac{1}{B^2} \sum_{0, i, j \leq B} \left(r_{i,j} - \hat{r}_{i,j} \right)^2 \qquad (3.16)$$

The algorithm of the proposed the proposed iteration-free fractal image coding using VQ is given as follows:

B. Encoder:

Step1: Partition the given image into range blocks X of size B x B and find the mean and variance of each X.
Step2: Plot the mean image using the mean of X as the pixel value and partition this mean image into blocks of size B x B and apply LBG algorithm to get the domain pool W of the required size N (N=16 / 32 / 64).

ISSN (Print) : 2319-5940
ISSN (Online) : 2278-1021

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 3, Issue 1, January 2014

Step3: Determine the domain pool's projection value on the first axis and arrange them in the ascending order of the projection values on the first axis. Determine the projection value c_{ij} and $|s_{ci}|$ (i=1, 2… N and j= 1, 2, 3) for all domain blocks in the domain pool. Construct the nearest distance table dt={d_{n1}, d_{n2}… d_{nN}}.
For each range block X:

Step4: If variance (X) < E assign 0 to label and μ_X to code. Process the next range block.
Step5: Assign 1 to label. Choose the domain block W_m and compute the distance between X and W_m, where W_m satisfies the following condition $|x_1 - c_{m1}| \leq |x_1 - c_{j1}|$, $1 \leq j \leq N$, and $j \neq m$. Let $r = d(X, W_m)$ and store the valve of r^2 to sqr. If $r \leq d_{nm}$, then W_m is the closest domain block of X. Go to step 11. Otherwise compute the projection values x_i (i=1, 2, 3) and $|s_x|$ of X. Set d=1.
Step6: If (m+d) $\geq$ N or the domain block W_{m+d} is deleted, go to step 8. Otherwise go to step 7.
Step7: a) Compute $D_i = |c_{(m+d)i} - x_i|$ (i=1,2,3).
If $D_1 \geq r$, then eliminate all domain blocks from W_{m+d} to W_n and go to step 8.

If $D_j \geq r$, (j=2,3) then delete domain blocks W_{m+d} and go to step 8.

b) Compute $D_t = \sum_{i=1}^{3} D_i^2 + (|s_{c(m+d)}| - |s_x|)^2$. If $D_t \geq$ sqr, then delete domain blocks W_{m+d} and go to step 8.
c) Compute r' = d (X, W_{m+d}) and set sqr' = (r')^2. If r' $\geq$ r then domain block W_{m+d} is eliminated and go to step 8. Otherwise set r = r' and sqr = sqr'. If $r \leq d_{n(m+d)}$, then W_{m+d} is the closest domain block of X, go to step 11. Otherwise go to step 8.

Step8: If (m-d) < 0 or the domain block W is deleted, go to step 10. Otherwise go to step 9.
Step9: a) Compute $D_i = |c_{(m-d)i} - x_i|$. i=1,2,3.
If $D_1 \geq r$, then eliminate all domain blocks from W_{m-d} to W_0

If $D_j \geq r$, (j=2,3) then delete domain blocks W_{m-d}

Go to step 10.

b) Compute $D_t = \sum_{i=1}^{3} D_i^2 + (|s_{c(m-d)}| - |s_x|)^2$. If $D_t \geq$ sqr, then delete domain blocks W_{m-d} and go to step 10.
c) Compute r' = d(X,W_{m-d}) and set sqr' = (r')^2. If r' $\geq$ r then delete domain block W_{m-d} , go to step 10. Otherwise set r = r' and sqr = sqr'. If $r \leq d_{n(m-d)}$, then W_{m-d} is the closest domain block of X, go to step 11 otherwise go to step 10.

Step10: Set d = d+1. If (m+d > N and m-d < 0) or (both W_{m+d} and W_{m-d} are deleted), go to step 11. Otherwise, go to step 6.
Step11: Apply the isometry transformations i to the minimized domain pool W for contrast scaling α = x/4 {for x=1 to 4}. Calculate the RMS error between the transformed domain blocks and the range block. Transfer the values of i, α, μ_X, index of W_j, which has the minimum RMS error to code. Process the next range block
The flow chart of the enhanced iteration-free fractal image coding using VQ is shown in fig. 5

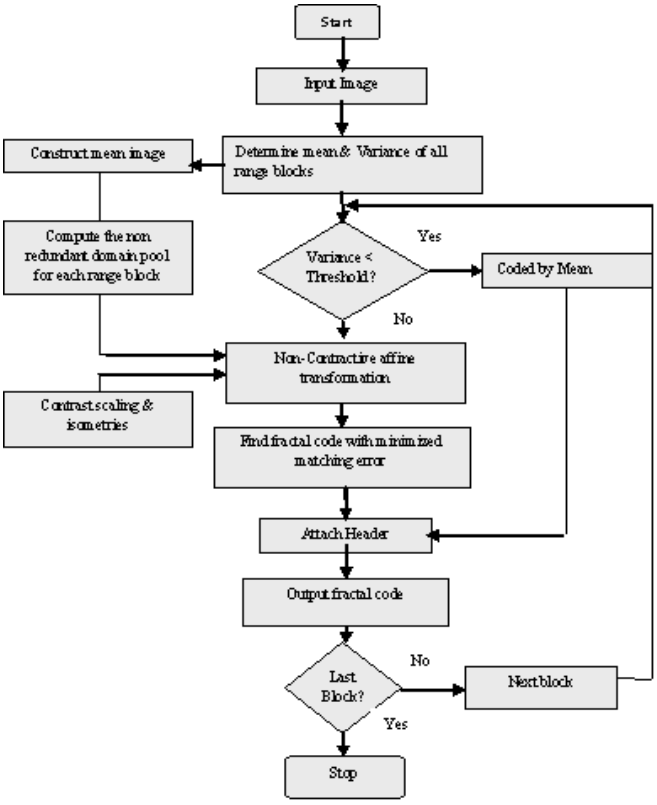


Fig. 5 Flow Chart of the Encoder for the Proposed Method Using VQ

IV. IMPLEMENTATION OF THE PROPOSED TECHNIQUES

These algorithms were implemented using the software Matlab 7.12 on the Intel (R) Core[TM]2 E7500 system with 2.93 GHz and 1.96 GB of RAM. For implementation of these algorithms, four 512 x 512 benchmark color images of Lena, Pepper, Cauliflower and TajMahal [shown in Figure 6 (a) to (d) with twenty four-bit RGB color resolution were used.

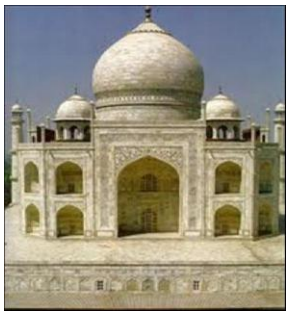
ISSN (Print) : 2319-5940
ISSN (Online) : 2278-1021

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 3, Issue 1, January 2014

(a)Lena

(b) Pepper

(c) Cauli Flower



(d)Taj Mahal

Fig.6. Original (512 X 512, 24 Bit/Pixel) Images.

The storage of an image within computer memory is dependent on the type of image and the required image size (the width and height are often powers of 2 for practical reasons) and resolution. Although pixels often take on floating point values during processing, unsigned integer valued pixels are standard for storage and display; an image with pixel values between 0 and 2 k - 1 requires k bits per pixel, and the image is said to be of depth k. Color images may be represented by three image planes (one each for red, green and blue) of 8 bits/pixel each, resulting in a total of 24 bits/pixel, or by 8 bits/pixel for an index into a color lookup table. It is important to emphasize that all of these quantities refer to memory allocation for images in uncompressed or canonical form, rather than the rate for coded images.

In order to obtain the same domain blocks in both the encoder and decoder in the fractal coding scheme, an iteration-free fractal coding scheme was proposed by [7]. This method was further improved in the proposed method by reducing the domain pool for each range block which results in efficient coding time.

The LBG algorithm was applied to design the domain pool using the mean image. The LBG method reduces the redundancies between the generated domain blocks and thus the constructed domain pool was efficient compared to the fractal schemes using iterations. The coding performance was further improved in the proposed algorithm. It used the vector features namely mean value, edge strength, and texture strength to delete impossible domain blocks for each range block. To find the better matching domain block for a range block, two additional conditions were executed. One condition helped in terminating the searching process and another to reject impossible domain blocks. This helped in the reduction of the distortion calculations for the computation of the better match for the range block. Extra computations for these additional conditions were very small and had reduced the coding time to a great extent. The computer simulation showed that the coding time was greatly reduced and the quality of the decoded image was also good. The domain pool design of three sizes were used in the proposed method namely N_D = 16, 32 and 64. The performance of the quality of the decoded image was evaluated by the root-mean-square-error and the coding time was computed.

The range block with a single size (8x8, 4x4 and 2x2) was considered for simulation. The length of the attached header to the proposed iteration-free fractal code for each range block was one bit because it only denoted whether or not the range block was coded by the mean. For an image partitioned by 4x4 range blocks, every block mean was calculated and a 128x128 mean image was obtained. Fig. 4.2 (b) shows the mean image of Lena got by this partition and it is very similar to its original image except its size. Therefore the domain pools of different sizes namely 16, 32 and 64 using the LBG-based method from the mean image was constructed. The coding performance with the contractive affine transformation under the different sizes for the domain pool on the parameters like coding time, image quality and bit rate was determined. For the image partitioned by 8x8,4x4 and 2x2 range blocks, the 64x64,128x128 and 256x256 mean images for Lena was obtained and shown in Fig. 7 (a),(b) and (c) respectively. The domain pools of different sizes were constructed on these images using the LBG-based method and computed the coding performance on the same parameters for different sizes of the domain pool.

ISSN (Print) : 2319-5940
ISSN (Online) : 2278-1021

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 3, Issue 1, January 2014

(a) 64x64 (b) 128x128 (c) 256x256

Fig. 7. Mean Images Of Lena.

V. RESULTS AND DISCUSSIONS

Experimental results of the coding time and image quality using different sizes of codebooks for the iteration-free fractal codes and the proposed enhanced iteration-free fractal method using VQ for single block size 8x8, 4x4 and 2x2 are tabulated in Table 1and 2. The proposed method provides better performances than the existing iteration-free fractal coding scheme in terms of coding time. The decompressed image of Lena using the block size 8x8, 4x4 and 2x2 using 64 levels are shown in the Fig. 8 (a) to (c) respectively. As the size of the mean image increases, the quality of the image becomes more nearer to the iteration-free fractal method. The RMS of the decoded image partitioned by the 8x8 block size is higher than that partitioned by the 4x4 and 2x2 block size since a smaller block size leads to a smaller matching error for the affine transformation. However, the bit rate increases significantly because the number of the 4x4 and 2x2 range blocks are four times and sixteen times the number of the 8x8 range blocks. The decompressed image of Pepper, Cauliflower, and Tajmahal for the block partition of sizes 8x8 using 16 levels, shown in the Fig. 9 (a) to (c) respectively.

(a) 2x2 (b)4x4 (c)8x8

Fig. 8. Decompressed Images of Lena.

Fig. 9. Decompressed Images of Pepper, Cauliflower and Tajmahal (8x8)

ISSN (Print) : 2319-5940
ISSN (Online) : 2278-1021

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 3, Issue 1, January 2014

TABLE I

CLASSIFICATION OF BLOCKS COMPRESSION RATIO, ENCODING TIME AND BIT RATE ON THE CHOSEN IMAGES USING THE PROPOSED TECHNIQUE

Image	Range Block Size	No of Range Blocks		16 level	32 level	64 level	Bit rate	Compression ratio
		Smooth	Rough	Encoding Time(sec)	Encoding Time(sec)	Encoding Time(sec)		
Lena	8x8	180	3916	5941.4	7009.6	7009.6	0.54	44.50
	4x4	2217	14167	23272	31386	31386	2.04	12
	2x2	22516	43026	79228	153450	153450	7.47	3.20
Pepper	8x8	355	3741	5855	6881	6881	0.53	45.19
	4x4	3474	12910	21268	28241	28241	1.99	12.06
	2x2	31705	33831	55990	109640	109640	7.16	3.35
Cauli flower	8x8	1320	2776	4562.8	5265.7	5265.7	0.49	48.87
	4x4	6437	9947	16334	22585	22585	1.87	12.77
	2x2	35214	30322	49001	96249	96249	7.04	3.40
Tajmahal	8x8	744	3552	8817.3	6408.1	6408.1	0.57	44.44
	4x4	3573	12811	21163	31149	31149	1.98	12.07
	2x2	26086	39450	70014	138210	138210	7.35	3.26

TABLE II
CLASSIFICATION OF BLOCKS PSNR AND RMS IMAGES USING THE PROPOSED TECHNIQUE

Image	Range Block Size	16 level		32 level		64 level	
		PSNR	RMS	PSNR	RMS	PSNR	RMS
Lena	8x8	32.78	5.85	32.73	5.89	32.78	5.85
	4x4	34.03	5.06	34.159	4.99	34.12	6.02
	2x2	35.81	4.13	35.81	4.13	35.88	4.09
pepper	8x8	32.49	6.05	32.47	6.06	32.46	6.07
	4x4	34.31	4.90	34.48	4.81	34.41	4.85
	2x2	36.80	3.68	36.893	3.65	36.964	3.61
cauliflower	8x8	32.12	6.31	32.08	6.34	32.05	6.36
	4x4	34.05	5.05	34.25	4.94	34.36	4.88
	2x2	38.04	3.19	38.19	3.14	38.40	3.06
Tajmahal	8x8	31.05	7.14	30.98	7.20	30.97	7.21
	4x4	32.83	5.82	32.93	5.75	32.92	5.75
	2x2	35.83	4.12	35.97	4.05	36.15	3.97

VI. CONCLUSION

Color images are commonly used in most of the application now-a-days. An RGB color image is an MxNx3 array of color pixels, where each color pixel is a triplet corresponding to the red, green and blue components of an RGB image at a specific spatial location. The fast encoding algorithm for iteration free fractal image coding is introduced. This algorithm uses three features namely mean value, edge strength, and texture strength of a vector to eliminate many of the unlikely domain blocks from the domain pool which is not available in the existing algorithm. From the experimental result, it concludes that when the bit rate is low, compression ratio. The bit rate is high and PSNR is very high. But the color image compression PSNR values are very high compared to the Grayscale images.

REFERENCES

[1] Erjun Zhao Dan Liu. *Fractal Image Compression Methods: A Review*. Proceedings of the Third International Conference on Information Technology and Applications (ICITA'05), 756-759, July 2005.

[2] M.Ghazel, R. K. Ward, R. Abugharbieh,E. R. Vrscay, G. H. Freeman. *Simultaneous Fractal Image Denoising And Interpolation*.0-7803-91950/IEEE, 558-561, 2005.

[3] Byung Cheol Song, Kang-Wook Chun and Jong Beom Ra. *A Rate-Constrained Fast Full-Search Algorithm Based on Block Sum Pyramid.* IEEE Trans. Image Processing, 14(3), pp. 308–311, Mar. 2005.

ISSN (Print) : 2319-5940
ISSN (Online) : 2278-1021

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 3, Issue 1, January 2014

[4] Chen Yisong; Wang Guoping; Dong Shihai. *Feature Difference Classification Method In Fractal Image Coding*. Proceedings of the 6th International Conference on Signal Processing, 1, 26-30 Aug. 2002, 648 – 651, 2002.

[5] Hau-Jie Liang and Shuenn-Shyang Wang. *Architectural Design Of Fractal Image Coder Based On Kick-Out Condition*.0-7803-8834-8/2005 IEEE, 1118-1121, 2005

[6] Riccardo Distasi, Michele Nappi, and Daniel Riccio. *A Range/Domain Approximation Error-Based Approach for Fractal Image*

[7] *Compression*, IEEE Transactions On Image Processing, 15(1), 89- 97**,** January 2006.

[8] Hsuan T. Chang and Chung J. Kuo. *Iteration-Free Fractal Image Coding Based on Efficient Domain Pool Design*. IEEE Trans. Image Processing, 9(3), 329-339, Mar 2000.

[9] A.R.Nadira Banu Kamal and S.ThamaraiSelvi,*Enhanced Iteration Free Fractal Image Coding Algorith, with efficient Search and Storage Space.*ICTACT Journal on Image and Video Processing,124-134, Nov 2010.

ACKNOWLEDGMENT

The work has been supported by Major Research Project, UGC, New Delhi.

BIOGRAPHIES

A R Nadira Banu Kamal, Professor of the Department of Computer Science of TBAK College for Women, Kilakarai, Ramanathaputam District, Tamil Nadu, India. I did my B Sc in Madras University, MSc in Anna University, MPhil in Alagappa University and PhD in Manonmanium Sundaranar University. My area of Interests and research are Digital Image Processing and Soft Computing. I am at present supervising 8 research scholars for their PhD degree and have guided many MPhil scholars.

P.Priyanga, Project Fellow for the UGC-Major Research Project, Department of Computer Science of TBAK College for Women, Kilakarai. Ramanathapuram District, TamilNadu, India. I did my B.Sc (C.S) in Government Arts College for Women, Ramanathapuram, M.Sc and M.Phil (C.S) in Sri Sarada College for Women, Salem. My area of Research and Interest are Digital Image Processing and Data Mining.