



ANALYSIS OF FRACTAL VIDEO CODING USING FIXED PARTITIONING SCHEME

KULKARNI M.V.¹ AND KULKARNI D.B.²

¹Jawaharlal Nehru Technological University, Hyderabad-500 085, AP, India.

²Department of Information Technology, Walchand College of Engineering, Sangli-416 415, MS, India.

*Corresponding Author: Email- kulk.mv@gmail.com

Received: July 12, 2012; Accepted: July 18, 2012

Abstract- A Fractal based video compression method relies on the concept of self similarity within a frame and between successive frames in a video. There are two fractal compression methods called Cube based fractal compression and Frame based fractal compression. We have adopted the frame based video compression method, where the first frame is encoded using the intra frame compression technique and subsequent frames are compressed using inter frame compression technique. For encoding each frame, range and domain blocks are generated by fixed block partitioning scheme. Fractal codes are computed for every frame by comparing domain blocks from previous frame with range blocks of current frame and are stored for decoding purpose. In the decoding process, frames are recreated by loading stored fractal codes. Experimentation is performed by considering different sizes of range and domain blocks. Results are compared for different block sizes on the basis of compression ratio and PSNR (Peak Signal to Noise Ratio). It is observed that least sized range blocks provide higher PSNR at the cost of lower compression ratio.

Keywords- Fractal, video compression, range block, domain block.

Citation: Kulkarni M.V. and Kulkarni D.B. (2012) Analysis of Fractal Video Coding Using Fixed Partitioning Scheme. Journal of Signal and Image Processing, ISSN: 0976-8882 & E-ISSN: 0976-8890, Volume 3, Issue 4, pp.-126-129.

Copyright: Copyright©2012 Kulkarni M.V. and Kulkarni D.B. This is an open-access article distributed under the terms of the Creative Commons Attribution License, which permits unrestricted use, distribution and reproduction in any medium, provided the original author and source are credited.

Introduction

Fractal coding is a new technique that has recently attracted the interest of researchers in the field of image and video coding. Amongst all new video compression methods, fractal video compression seems to be a favorable method amongst many researchers [1, 9]. It decodes an image by representing the regions as geometrically transformed versions of other regions in the same image.

The basic principle of fractal based image compression methods is to find out or approximate each block of the image by assigning contractive transformation on some larger blocks in the image [6].

Fractal image or video compression is a compression method which is based on self-similarity within the different portions of the image. Fractal Compression technique has the property that it creates artificial detail at every scale. This is the reason why the method is resolution independent. It also achieves a higher compression ratio [1, 3].

Consider an arbitrary gray scale image, P , of size $J \times J$ pixels. This image may be partitioned in two different ways: Range blocks and Domain blocks. The range blocks, R are a set of non-overlapping

image blocks of size $k = n \times n$, which are denoted as $\{R_i\}_{i=1}^{nr}$.

The number of range blocks is given by, $nr = \frac{1}{n} \times \frac{1}{n}$. The original

$$P = \bigcup_{i=1}^{nr} R_i$$

image P is a union of

The domain blocks are also sub partitions of the original image P and they must cover the whole image. Unlike the range blocks, the domain blocks may be overlapping and are larger in size than a range block for compressing the image. One way in which the domain blocks may be obtained is by sliding a window of size $l = m \times n$, where $m > n$, throughout the image to construct the domain pool [4].

To encode a range block R , each of the blocks in the domain pool is scaled to the size of the range block. Then it is compared to R with respect to intensity offset and contrast parameters, as well as the eight isometric transformations (the identity, reflections about the mid-horizontal, mid-vertical axes, reflections about each diagonal, and rotations through 90, 180 and 270). The set of contracted

domain blocks is denoted as $\{D_i\}_{i=1}^{nd}$ where nd is the number of

domain blocks in the domain pool. Each domain block has to be scaled and the eight transformations must then be applied. The resulting pool of size $8 \times n_d$ is called a code book pool. The domain block which has the closest match with R from the codebook pool is selected as the best matched block [4].

Since the overall transformation is a collection of all the transformations on the domain block, it is given by [4]:

$$P = W(P) = \bigcup_{i=1}^n w_i(D_i)$$

In the case of grayscale images, (w_i) affine transformations take the form

$$w_i \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} a_i & b_i & o \\ c_i & d_i & o \\ 0 & 0 & s_i \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} + \begin{bmatrix} e_i \\ f_i \\ o_i \end{bmatrix}$$

where s_i controls the contrast and o_i controls the brightness of the transformation. The specific choice of parameters for the affine transformation w_i are determined by minimizing the following quantity [4, 6]:

$$R_{MSE} = R - \|s(D + oI)\| \tag{1}$$

Here I denote the Identity matrix of dimension N, s and o are the contrast and offset parameters respectively, which must be determined in advance to calculate the spacing between domain D and range R. The contrast factor should ensure the contractility of the

transformation. The metric $R_{MSE} = R - \|s(D + oI)\|$ is the mean square error (MSE) metric. This metric expresses the distance between two images or two range blocks [1,4].

The objective when determining contrast and brightness is to minimize the Root Mean Square Error (RMSE). Contrast is calculated by following formula:

$$s = \frac{N \left(\sum_{i=1}^N D_i R_i \right) - \left(\sum_{i=1}^N D_i \right) \left(\sum_{i=1}^N R_i \right)}{N \left(\sum_{j=1}^N D_i^2 \right) \left(\sum_{j=1}^N D_i \right)} \tag{2}$$

Brightness is calculated by following formula:

$$o = \frac{\left(\sum_{i=1}^N R_i \right) - s \left(\sum_{i=1}^N D_i \right)}{N} \tag{3}$$

By substituting (2) and (3) into (1), the distance R_{MSE} can be computed as (4):

$$R_{MSE} = \frac{\left(\sum_{i=1}^N R_i^2 \right) + s \left(s \left(\sum_{i=1}^N D_i^2 \right) - 2 \left(\sum_{i=1}^N D_i R_i \right) + 2o \left(\sum_{i=1}^N D_i \right) \right) + o \left(No - 2 \left(\sum_{i=1}^N R_i \right) \right)}{N}$$

The domain block which results in the smallest value of RMS error is then selected as the best matched block, and the related parameters for the transformations may be encoded and stored. The smallest value of RMS error indicates that the best matched domain block is found [1, 4].

System Design

The block diagram for fractal encoding is shown in [Fig-1] given below. In the process of Fractal encoding, the input video is converted into images. The images are partitioned into smaller sections called range blocks and domain blocks which are of size twice that of the range blocks [7]. Any one of the partitioning techniques i.e. Fixed block, Quad tree, H-V or Triangular partitioning can be used.

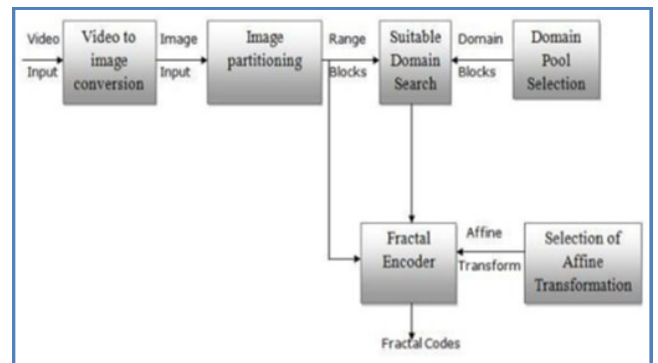


Fig. 1- Process-Cycle Encoding

In this experimentation, the fixed block partitioning technique is used. Eight types of affine transformations are performed which include rotation by 0, 90,180 & 270 degrees, horizontal & vertical flip and forward and reverse diagonal flip. Compression codes are computed for each image by comparing it with the previous frame. The same process is repeated for all images in the video.

The block diagram for fractal decoding is shown in [Fig-2] given below. The input for the fractal decoder is the file of fractal codes generated by the fractal decoder.

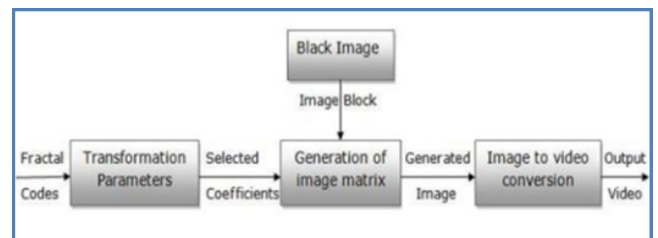


Fig. 2- Process-Cycle Decoding

A black image is created as a starting image. Coefficients are selected from the fractal codes file and applied on the initial image. An image matrix is generated by repeating this process for each set of blocks in the initial image. The same process is repeated till the last image is generated. Using image to video convertor, the same video is generated in the compressed format.

Encoding Algorithm

The encoding process of fractal image compression is computationally intensive. In case of fractal video coding, the first frame is encoded using fractal coding and for the successive frames, previ-

ous frame is used as a reference frame or the source of domain blocks [8, 10]. The following algorithm indicates the flow of the encoding process.

1. Accept a video as input.
2. Convert it into frames.
3. Let n=number of frames
4. For i = 1 : n
 - Consider first frame as the reference frame.
 - Compare ith frame with (i + 1)th frame.
 - Determine the 2nd order geometric transformation (skew, stretch, rotate, scale and translate) for each object in successive frames.
 - Determine the difference between pixel values in successive frames.
 - If difference in pixel values in successive frames is less than threshold then store transformation values of the later in terms of previous (reference) frame.
 - Otherwise consider current frame as reference frame
5. End

Decoding Algorithm

The decoding in fractal compression is much faster compared with encoding; here the time depends on the number of iterations. Usually 4 to 8 iterations are required to reach the fixed point or the attractor in fractal image/video compression. Following are the steps carried out in the decompression process of fractal-based compression:

1. Load the saved coefficients.
2. Allocate memory buffers for the range and domain images.
3. Apply the Affine coefficients on the domain image.
4. Copy the content of the domain block to the range block.
5. Repeat the procedure till the last frame is created.
6. Regenerate the video by accumulating all the frames as the same sequence as they were in the original video.

Experimentation and Results

Experimentation is performed using MATLAB 7.0 and the programs are executed on PC with CPU speed equal to 1.73 GHz. Following formulae are used to determine various parameters for comparing the results. The generalized formula for compression ratio is given by:

$$\text{Compression ratio} = \frac{\text{Size of uncompressed image}}{\text{Size of compressed image}}$$

The PSNR is the most commonly used as a quality measure of reconstruction of lossy compression codes. Let I and K be two monochrome images with size m x n. The mean square error (MSE) is calculated as

$$MSE = \frac{1}{mn} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} [I(i, j) - K(i, j)]^2$$

The PSNR is defined as

$$PSNR = 10 \cdot \log_{10} \left(\frac{MAX_I^2}{MSE} \right) = 20 \cdot \log_{10} \left(\frac{MAX_I}{\sqrt{MSE}} \right)$$

Here, MAX_I is the maximum pixel value of the image. When pixels are represented with b bits per sample, MAX_I is 2^b - 1.

Table 1- Decoding

Iterations	Range Size = 2	Range Size = 4	Range Size = 8	Range Size = 16
1	25.51	21.62	19	16.85
2	27.25	24.6	21.02	18.35
3	27.96	26.52	21.92	18.81
PSNR in dB	27.89	25.77	22.06	18.87
4	27.61	25.86	21.98	18.86
5	27.76	25.9	21.99	18.86
6	27.87	25.9	21.98	18.86
7	27.87	25.88	21.97	18.86
8				

[Table-1] indicates the performance of Fractal Compression algorithm for the Lenna gray test image with size 128 x 128. Comparison is based on the number of iterations used in the decoding process. The same image is applied for fractal compression with size of range blocks as 2, 4, 8 and 16. The quality of the generated image is tested by calculating PSNR values.

These results in graphical format are shown below in [Fig-3].

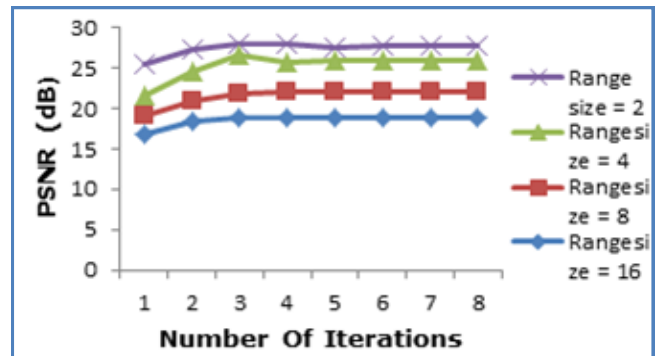


Fig. 3- Decoding (Number of Iterations vs. PSNR)

It is observed from the [Fig-3] that the PSNR value is low for lesser number of iterations and it increases with increase in the number of iterations. After each iteration the output image is fed back to the decoder to reduce the error and the process is repeated till desired attractor (final image) is reached. It can also be observed that after approximately 4 iterations, the PSNR value remains constant which means that the desired attractor is reached.

Peak Signal to Noise Ratio

It measures the amount of noise in the regenerated video. The algorithm is tested for different types of videos and it is observed that maximum PSNR is achieved at less size of range blocks as shown in the [Table-2]. It is so because with higher sized range blocks, more image area is covered and hence during encoding process, error is introduced in coefficient determination. This error affects the regenerated image and thus resultant image is merely an approximate representation of the original image and hence original video. But with lesser sized range blocks, lesser image area is covered and size of domain blocks is also less and hence error is minimized in coefficient determination procedure. These results in graphical format are shown below in [Fig-4].

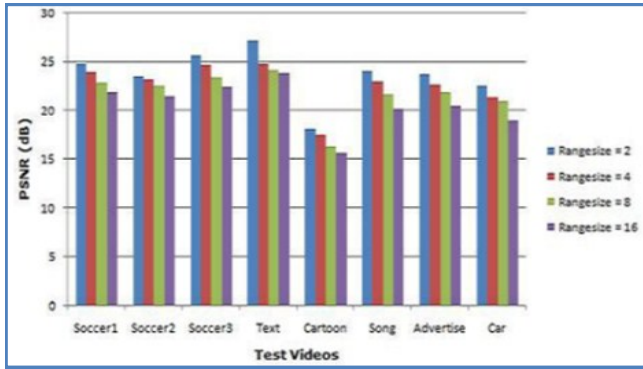


Fig. 4- Decoding of Various Videos (Videos vs PSNR)

Compression Ratio

The [Table-2] shows a comparison for different range sizes for various test videos. It can be observed that for least sized range blocks, compression ratio is lesser and as range size is increased, compression ratio also increases. This behavior is observed as for larger range sizes, the amount of detail in reconstructed video is lesser than those with lesser range sizes. For range size=2, the reconstructed video is almost similar to the original video and hence it is not compressed to the same extent as it is with range size=16.

Table 2- Analysis of Compression Ratio

Video	Compression Ratio			
	Range size = 2	Range size = 4	Range size = 8	Range size = 16
Soccer1	2.68	4.48	5.97	7.27
Soccer2	3.87	5.12	5.56	6.8
Soccer3	2.91	3.85	4.63	6.32
Text	1.46	2.23	3.72	5.08
Cartoon	2.47	3.49	4.06	5.49
Song	3.24	4.22	5.03	6.31
Advertise	3.06	3.96	4.67	5.82
Car	3.17	4.88	6.56	8.7

Advantages and Disadvantages of Fractal Compression

When the fractal video compression is compared to other methods used to compress different videos, some of the main advantages and disadvantages can be summarized as given below.

Advantages of Fractal Compression

- Good mathematical encoding frame.
- Resolution independence.
- High compression ratio.
- Decompression is fast.

Disadvantage of Fractal Compression

- Computationally Intensive

Conclusion

Image and Video Compression: It is necessary to reduce the space required for storing data and also to reduce the time required for transmission of data. With lossless compression we cannot achieve high compression ratio but the decoded image or video will be exactly same as the input; on the other hand with lossy type, we obtain high compression rates but some data is lost. However

that loss of data is not significant since human visual systems cannot detect it.

Fractal Compression: It falls under the category of lossy compression. It works well for videos with steady background instead of continuously changing background. It gives best results when used for natural images and its worst results are found for videos containing only textual data. Fractal Video Compression is performed in the same way as Fractal Image Compression is achieved. The major difference between them is that, video compression takes advantage of the similarity between successive frames and hence we obtain higher compression rates compared with image compression.

Experimentation and Results: From the experimentation and results, it can be concluded that for less size range block (2 x 2), we get a lower compression ratio but higher quality decompressed image. As the size of range blocks is increased, i.e. (4 x 4), (8 x 8), (16 x 16), good compression ratio is observed but quality of the image degrades drastically. It can also be observed that we get higher PSNR for smaller range blocks than larger sized domain blocks.

Future Work: Lack of Speed is the major drawback of fractal compression technique. But it can be handled with the help of various approaches like grouping of domain blocks, and use of higher configuration machines.

Acknowledgement: We express our thanks to all the authors, those who are mentioned in the references, about their contribution in Video Coding and fractals which will enable us to carry out this experimentation.

References

- [1] Yuval Fisher (1995) *Fractal Image Compression: Theory and Application*, Springer-Verlag Publication.
- [2] Meiqing Wang, Choi-Hong Lai (2005) *Comput. Math. Appl.*, 50 (3-4), 611621.
- [3] Andonova S., Popovic D. (1994) *Digital Object Identifier*, 1, 343-348.
- [4] Dan Liu and Peter K. Jimack (2007) *Journal of Algorithms and Computational Technology*, 1, 171-186.
- [5] Fakhraldeen H. Ali, Azzam E. Mahmood (2006) *Al-Rafidain Engineering*, 14(4).
- [6] Arnaud E. Jacquin (1990) *SPIE*, 1360, 227-239.
- [7] Saupe D., Hamzaom R. and Hartenstem H. (1996) *Fractal image compression--An introductory overview*, In *Fractal Models for Image Synthesis*, Compresswn and Analyzs, ACID SIGGRAPH'96 Course Notes 27.
- [8] Luis Torres, Murat Kunt (1996) *The Second Generation Approach*, Kluwer Academic Publishers.
- [9] Mandelbrot B.B. (1982) *The Fractal Geometry of Nature*. W.H. Freeman and Company.
- [10] Al-Asmari K. (2003) *Int. J. Network Manage.* 13, 3-10.