Speeding-up Fractal Audio Compression Using Moment Descriptors

Azhaar K. Bedan[1], Loay E.George[2]
Computer of Science, College of Science, University of Baghdad, Iraq

Abstract— The domain-range affine mapping step of the fractal encoding process is the core step and the most costly one; it is inhibitive due to the large number of involved calculations. This research aims to filter out the domain blocks which have no chance of being selected as the best approximation for a given range block and thus reduce the number of domain-range comparisons in order to reduce the long encoding time which is considered as the main drawback of the fractal audio compression. The speeding up of the encoding process is achieved by adopting a selective based search strategy instead of the traditional search approach. The proposed approach based on using block indexing method to classify the domain pool blocks. The block indexing method utilizes the moment descriptors as indexing parameters to group the domain pool into classes each of which has a specific classifier; thus instead of making the exhaustive search through the whole domain blocks, only a subset of these blocks are tested. The tested blocks affiliated to one of the domain pool classes that have the same descriptors as the tested range block. The tests conducted on different audio samples showed high reduction in the encoding time in comparison with that for traditional method without significantly affect the quality of the reconstructed audio file.

Index Terms— Fractal audio compression, Moment descriptor, Block indexing, Selective search.

1 INTRODUCTION

Compression of audio signals has found applications in many areas, such as multimedia signal coding, high fidelity audio for radio broadcasting and audio transmission for HDTV, audio data transmission/sharing through the internet etc [1]. The need for audio compression algorithms that can satisfy the conflicting demands of high compression ratios and transparent quality for high fidelity audio signals led to the establishment of several coding methodologies over the last two decades [2]. Among the audio compression techniques, the fractal audio coding method represents one of the most important and prevailing audio compression techniques which are of current interest [1].

Since the pioneer work of Barnsly [3], followed by Arnode Jacquine [4] about the use of iterated function system (IFS) for image compression, many fractal compression techniques had been growing rapidly during the past two decades. Fractal compression is mainly studied on images. Many researchers have studied and improved the fractal image encoding and have gotten a lot of achievements [5]. The greater success of this method when applied to images encouraged several researchers to explore its applicability on other types of data (such as on audio data) [6]. So, the fractal audio compression can be considered as an advance step in the life cycle of the fractal coding. However, the earlier researches in fractal based audio compression have left a great bruit about this method, primarily its long encoding time even for small audio files. Some of the earlier researches integrated the fractal coding into the framework of conventional compression techniques and transform methods such as wavelet transform [7]. These researches based on the assumption states that the effectiveness of fractal image compression is due to its ability to approximate discontinuous functions. As such, they are not suitable for the approximation of audio signals which exhibits greater smoothness; in other words the fractal coding is applied indirectly to audio signals.

In this research, the fractal coding is applied directly on audio signals with a great reduction in the encoding time. This is achieved by exploiting two different sets of centralized moments to determine two different descriptors for each range/domain block. These descriptors are used individually to implement fast fractal audio compression systems.

- *Azharr K. Bedan has master Degree in Computer science in University of Baghdad, Iraq. E-mail: azhaar78_kh@yahoo.com.*
- *Loay E. George has PH.D degree in Computer science in University of Baghdad, Iraq. E-mail: loayedwar57@yahoo.com.*

2 THE PROPOSED APPROACH

The two adopted concepts that play the major role in introducing a novel scheme to speed up the audio encoding process are: moment descriptors and block indexing method. These two concepts have been utilized to devise a new search strategy known as selective based search strategy. According to this scheme, each range block, which is characterized by its descriptor value, is matched only with those blocks have similar descriptor value. This will reduce the burden of IFS matching trails by reducing the number of domain blocks nominated to be the affine approximate for the tested range block.

2.1 Moment Descriptor

In general, moments are set of parameters which describe the distribution of material from a reference point or an axis. The idea of using moments to construct the audio feature vector is, today, one of the most commonly approach to discriminate blocks. Moments have different orders each reflects different information for the same audio [8].

In this research work, a set of centralized moments is utilized to determine the descriptors for the blocks belong to the domain and range pools. The used moments are of first (m_1 and $\bar{m}_1$) and third (m_3) orders, the odd orders are selected due to their sensitivity to block reflection. For an audio block, I(), these centralized moments are determined using the following mathematical representations:

$$m_1 = \sum_{i=0}^{L-1} (i-c)\, I(i) \qquad (1)$$

$$\bar{m}_1 = \sum_{i=0}^{L-1} w(i)\, I(i) \qquad (2)$$

$$m_3 = \sum_{i=0}^{L-1} (i-c)^3\, I(i) \qquad (3)$$

Where,

$$w(i) = \begin{cases} i & if\ i < c \\ -L + i + 0.5 & if\ i \geq c \end{cases} \qquad (4)$$

and,

$$c = \frac{1}{2}\,(L-1) \qquad (5)$$

L is the number of samples in each audio block $I(i)$.
$I(i)$ is the i^{th} sample in the audio block.

Either one of the two descriptors (F_1) and (F_2) for domain and range blocks are determined using the following mathematical expressions:

$$F_1 = \frac{m_1^2 - \bar{m}_1^2}{m_1^2 + \bar{m}_1^2} \qquad (6)$$

$$F_2 = \frac{m_1^2 - m_3^2}{m_1^2 + m_3^2} \qquad (7)$$

In this research, the two above descriptors are used individually to implement two different classifications for the domain pool blocks. Thus, during the encoding process each range block is matched with a particular class of domain blocks which is selected from the classes that are either constructed based on using the first type descriptor (F_1) or from the classes that are grouped according to the second type descriptor (F_2).

2.2 Block Indexing

In general, indexing is used to speed up the access time to the desired data. This concept is utilized in this research to speed up the access to a desired set of domain blocks which some of its members represent the best approximation for a particular range block. The application of this concept requires building a data structure to guarantee faster access to the data. This data structure may contain the original data record with an index number for that record.

To determine the index values for each range and domain blocks, the determined descriptor values (F_1 and F_2) are mapped into integer values using the following equations:

Where,

$$Index = round(|F| * NoBins) \qquad (8)$$

F represents either F_1 or F_2 (depending on the adopted type of the two proposed descriptors.

NoBins represents the number of indexing bins, or in other words represents the range of the indexing values (i.e. [*0, NoBins*]).

3 Encoding Process

Before going into the details of the encoding process that adopts the developed selective based search strategy, the layout of this process is depicted first in Figure (1).

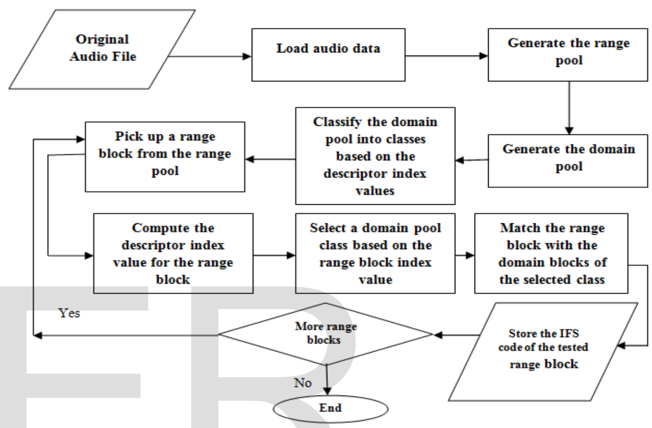

Fig. 1 The layout of encoding process

To implement the selective based search strategy the following block indexing algorithm had been implemented:

1. Load WAVE audio file and put its data in a one dimensional array.
2. Construct the domain and range pools.
3. For each domain block in the domain pool
   a. Determine the centralized moments (m_1, $\bar{m}_1$ and m_3) for the picked domain block using the equations: (1), (2) and (3).
   b. Determine one of the two descriptors (F_1) or (F_2) using equations (6) or (7) respectively.
   c. Determine the moment index value (*max*) for (F_1) or (F_2) using equation (8).
   d. Register the position of the domain block (Pos) and its moment index value (*max*) in their corresponding position in a temporary array (First_D for F_1 or Second_D for F_2).
   e. Sort the records of any one of the above mentioned arrays in ascending order according to the moment index values.
   f. Establish a set of pointers (P) to determine the boundaries of each group of blocks that have the same descriptor index values.
4. For each range block listed in the range pool:
   a. Determine the moment descriptor value then determine its moment index value.

b. Use the set of pointers (P) to determine the start and end of the set of domain blocks (class) that have the same index values as the tested range block.

c. If the selected group of the domain blocks (class) has sufficient number of blocks then among them the best approximation for the tested range block can be found among domain blocks of the selected class. In each matching instance determine the scale (S) and error (x^2) using the following equations [9]:

Where,

$$\acute{r}_i = s(d_i - \overline{d}) + \overline{r} \quad (9)$$

$$s = \begin{cases} \frac{\frac{1}{n}\sum_{i=0}^{n-1} d_i r_i - \bar{r}\bar{d}}{\sigma_d^2} & if\ \sigma_d^2 > 0 \\ 0 & if\ \sigma_d^2 = 0 \end{cases} \quad (10)$$

$$\chi^2 = \sigma_r^2 + s\left[s\sigma_r^2 + 2\overline{d}\,\overline{r} - \frac{2}{n}\sum_{i=0}^{n-1} d_i r_i\right] \quad (11)$$

(r_i') is the approximation value of the i^{th} sample of the range block.

d_i is the value of the corresponding i^{th} sample in the tested domain block.

$\bar{r}$, $\bar{d}$ are the average values of the tested range and domain blocks.

σ_d^2, σ_r^2 are the variance values of the domain and range blocks.

$$\overline{r} = \frac{1}{n}\sum_{i=0}^{n-1} r_i \quad (12)$$

$$\overline{d} = \frac{1}{n}\sum_{i=0}^{n-1} d_i \quad (13)$$

$$\sigma_d^2 = \frac{1}{n}\sum_{i=0}^{n-1} d_i^2 - \overline{d}^2 \quad (14)$$

$$\sigma_r^2 = \frac{1}{n}\sum_{i=0}^{n-1} r_i^2 - \overline{r}^2 \quad (15)$$

d. Compare the computed error value (x^2) of each matching instance with the minimum error (x^2_{min}) of the previous matching instances; such that if (x^2) is smaller than (x^2_{min}) then set ($x^2_{min} = x^2$), and register the scale value (S) and the position (Pos) of the matched domain block as the best IFS matching parameters.

e. Compare (x^2_{min}) with a predefined permissible error value (E_{thr}), if ($x^2_{min} < E_{thr}$) then the search is stopped and the set (S, Pos, r) is output as the best IFS code for the tested range block, then go to step 4g.

f. Otherwise if the selected class has insufficient number of blocks to get an acceptable approximate for the tested range block, then use a search window (denoted F_Window) to control the number of the neighboring classes that should be involved in the matching process to find the near optimal code (IFS code) for the tested range block.

Store the IFS code for the tested range block in the compression stream and then return to the beginning of step 4.

4 Test Results

This section is dedicated to present the results of the conducted tests to study the effects of some involved coding parameters on the performance of the developed FAC scheme. To study the effects of one of the coding parameters, the values of other parameters have been fixed. The proposed system was established using Visual Basic (Version 6) as SW development tool, and the HW environment used during our test is a single PC with Intel ® Core TM (2.53 GHz), and 4.00 GB RAM. In all these tests 8 bit mono waves of different size are used as test materials.

3.1 Number of Bins Tests

This set of tests was performed to investigate the effects of number of bins (NoBins) parameter on the compression performance parameters of the developed FAC scheme. In this test a wave file of size 266 KB is used as test materials.

Table (1) Effects of NoBins on the compression performance of the developed approach

Descriptor Type	NoBins	PSNR	CR	ET
F_1	10	45.215	3	61.621
	30	44.475	3	31.745
	40	44.039	3	18.314
	55	43.688	3	13.929
	70	43.424	3	11.358
	85	43.190	3	9.890
	100	43.009	3	8.441
F_2	10	42.223	3	110.214
	30	40.015	3	52.402
	40	39.639	3	31.683
	55	39.347	3	23.586
	70	39.154	3	18.831
	85	38.999	3	15.803
	100	38.892	3	13.541

The results show that PSNR and the encoding time (ET) are inversely proportional with NoBins, while the compression ratio (CR) is not affected by NoBins variation.

3.2 The Minimum Permissible Error Value (E_{THR})

This parameter is used to stop the matching trails of the domain blocks that belong to a particular class. So, if the registered error during any IFS based matching instance between a pair of domain and range blocks is less than (E_{thr}) then the search process is stopped and the current matching instance is chosen as the best matched domain. In this set of tests a wave file of size 266 KB is used as testing material. In this set of tests the value of NoBins Parameter is set to 30.

Table (2) Effects of E_{thr} parameter on the compression performance of the developed approach

Descriptor Type	E_{thr}	PSNR	CR	ET
F_1	1	44.475	3	31.745
	2	44.328	3	23.543
	3	44.136	3	18.513
	4	43.866	3	14.682
	5	43.514	3	11.825
F_2	1	40.015	3	52.402
	2	39.977	3	41.495
	3	39.923	3	35.741
	4	39.849	3	30.467
	5	39.746	3	26.722

This set of tests show that E_{thr} is inversely proportional with both the PSNR and the encoding time (ET).

3.3 F_WINDOW SIZE TESTS

This set of tests was performed to study the effects of F_WindowSize parameter on the compression performance parameters of the developed FAC schemes. As mentioned earlier the size of the search window controls the number of the neighboring classes whose domain blocks are considered as part of test domain when the class that associated with the index value of the tested range block has insufficient blocks to find the best approximate. In this set of test a wave file of size 38.1 KB is used as the testing material. During this test the values of other coding parameters are set: E_{thr}=2, NoBins=100.

Table (3) Effects of F_WindowSize parameter on the compression performance parameters of the developed approach

Descriptor Type	F_Widow Size	PSNR	CR	ET
F_1	1	33.594	3.310	0.344
	2	33.739	3.310	0.422
	3	33.859	3.310	0.499
	4	33.904	3.310	0.559
	5	33.932	3.310	0.639
F_2	1	32.690	3.310	0.406
	2	33.105	3.310	0.453
	3	33.256	3.310	0.499
	4	33.420	3.310	0.594
	5	33.527	3.310	0.610

This set of test results show that both PSNR and ET are directly proportional with the F_WindowSize parameter.

3.4 COMPARISON WITH RESULTS OF THE TRADITIONAL APPROACH

This set of tests was conducted to compare the performance of our developed FAC scheme (which adopts a selective based search strategy) with the performance of the traditional approach in which an exhaustive search is made through the whole domain pool in order to find the near optimal code for each block in the range pool. In these tests, four wave samples of different size have used as testing materials. The values of the coding parameters of our developed FAC system are set as: NoBins=100, F_WindowSize =1, E_{thr}=2.

Table (4) Comparison between Results of Our Developed Scheme and The Traditional Approach

Testing Samples	Traditional Approach		First Descriptor Approach			Second Descriptor Approach		
	PSNR	ET	PSNR	ET	RTR	PSNR	ET	RTR
Sample1 (38.1 KB)	38.216	36.005	33.594	0.344	104.6	32.690	0.406	88.6
Sample2 (61.7 KB)	42.545	94.427	38.688	0.736	128.3	37.186	1.062	88.9
Sample3 (99.7 KB)	46.140	245.748	42.234	1.529	160.7	38.897	2.389	102.8
Sample4 (266 KB)	46.029	1629.781	43.009	8.441	193.1	38.892	13.541	130.3

Where RTR stands for Reduction Time Ratio which is determined using the following equation:

$$\text{Reduction Time Ratio } (RTR) = \frac{\text{Time of traditional encoding method}}{\text{Time of the proposed method}}$$

4. CONCLUSIONS

The results of the conducted tests show that the encoding process based on the first descriptor (F_1) led to better encoding features (in terms of the elapsed encoding time and the quality of the reconstructed audio file) than that depends on the second descriptor (F_2). But the most important result presented in this research and which can be considered as a quality step in the scope of studying fractal audio compression is the high reduction in the encoding time obtained of the PIFS without making significant degradation in the audio quality.

Some improvements can be added to the developed scheme such as using the tiling concept in order to partition the input wave files into tiles; each tile is then submitted individually to the encoding process.

REFERENCES

[1] Gunjal, S. D. and Raut, R. D. ; "Advance Source Coding Techniques for Audio/Speech Signal: A Survey"; International Journal of Computer Technology and Applications; Vol. 3, No.4; pp. 1335-1342; 2012.

[2] Spanias, A., Painter, T., and Atti, V.; "Audio Signal Processing and Coding"; WILEY-INTERSCIENCE; 2007.

[3] Barnsley, M. F.; "Fractals Everywhere"; Academic Press; 1993.

[4] Jacquin, A. E.; "Image Coding Based on a Fractal Theory of Iterated Contractive Image Transformation"; IEEE image processing; Vol. 1, No. 1; pp. 18-30.

[5] Nodehi, A., Mohamadi, H. and Tayarani, M.; "A Distributed Quantum Evolutionary Algorithm With a New Cycling Operator and Its Application in Fractal Image Compression"; International Journal of Artificial Intelligence & Applications (IJAIA); Vol. 3, No. 1; pp. 163-176; 2012.

[6] Xiao, H. ; "Fractal Audio Coding"; M.Sc. Thesis; School of Computing; Queens University; Canada; 2005.
[7] Wannamaker, R. A. and Vrscay, E. R.; "Fractal Wavelet Compression of Audio Signals"; Department of Applied Mathematics; University of Waterloo, Canada; Internet Paper; 2009.
[8] George, L. E. and Al-Hillo, E. A.; "Speeding Up Fractal Color Image Compression Using Moment Features Based on Symmetry Predictor"; Eighth International Conference on Information Technology: New Generations; 2011.
[9] George, L. E.; "IFS Coding for Zero-Mean Image Blocks"; Iraqi Journal of Science; Vol. 47, No. 1; pp. 190-194; 2006.