Periodica Polytechnica Electrical Engineering and Computer Science, 64(3), pp. 303–312, 2020

Investigating the Effect of Block Length on the Performance of Fractal Coding Using Audio Files

Ahmed Hussain Ali[1]*, Loay Edwar George[2], Omar S. Saleh[3], Mohd Rosmadi Mokhtar[4], Qusay Al-Maatouk[5]

[1] Ministry of Higher Education and Scientific Research, Studies Planning and Follow-up Directorate, 10065 Al Tarbawi Collection, Baghdad, Iraq
[2] College of Science, University of Baghdad, 10070 Al-Jadriya, Baghdad, Iraq
[3] School of Computing, University Utara Malaysia, 06010 Kedah, Malaysia
[4] Centre of Cybersecurity, Faculty of Information Science and Technology, Universiti Kebangsaan Malaysia, 43600 Bangi, Selangor, Jalan Bunga Raya, Malaysia
[5] School of Technology, Asia Pacific University of Technology and Innovation, 57000 Kuala Lumpur, Jalan Teknologi 5, Malaysia
* Corresponding author, e-mail: ahmedhussainali78@gmail.com

Received: 26 August 2019, Accepted: 08 January 2020, Published online: 17 March 2020

Abstract

The goal of compression techniques is to reducing the size of data and decreasing the communication cost while transferring data. Fractal based coding technique is widely used to compress images files which provides high compression ratio and good image quality. However, like a compression technique, it is still limited because of the difference of the human perceptions between audio and image files, the long time for searching the best possible domain blocks and many comparisons in the encoding process. For those reasons, Fractal Coding had not broadly studied on audio data. Few years ago, Fractal Coding has been extended to apply on the audio data. In this paper, the application of the Fractal Coding on different types of audio files is investigated. Moreover, the effect of block length on the audio quality and compression performance are highlighted since block length is considered the main factor in the Fractal Coding algorithm. A GTZAN dataset is adopted in the evaluation and the experimental results show that there is an inverse relationship between block length and audio quality and proportional relationship between block length and compression ratio and factor. Furthermore, it can be noticed that the Fractal Coding can be compressed any speech and music audio signal directly with acceptable quality, PSNR 39 dB on average with a high compression ratio around 90 % with compression factor around 10 when the block length is 20 samples.

Keywords

audio compression, audio quality, block length, compression ratio, Fractal Audio Coding

1 Introduction

Data compression algorithms are used to reduce the number of bits required to represent any file such as an image, video, or music. It is also can be defined as the art or science of representing information in a compact form. The reason behind the need for data compression is that there are more information is generated and used in a digital form consisting of numbers represented by bytes of data [1].

Two algorithms are used to compression techniques: compression, and reconstruction. A compression algorithm takes an input and generates a compressed file that requires fewer bits while a reconstruction algorithm operates on the compressed file to generate the reconstructed file. The compression and reconstruction algorithms together are called the compression algorithm. Based on the requirements of a reconstruction file, data compression schemes can be divided into two classes as discussed in details in [2, 3] which are lossless and lossy compression schemes. Huffman Coding, Run Length Encoding, and Arithmetic Encoding are common examples of lossless compression techniques [4] while Discrete Cosine Transform (DCT), Discrete Wavelet Transform (DWT), Vector Quantization (VQ), and Fractal Coding (FC) are examples of lossy compression [1, 5–7].

Regarding audio coding, FC is applied in early of 19' the last century on image files and after that on audio files. However, there are many doubts about application FC alone on the audio files [8]. This conclusion is refuted after 2007 as shown in Section 3 by applying FC on audio

Cite this article as: Ali, A. H., George, L. E., Saleh, O. S., Mokhtar, M. R., Al-Maatouk, Q. "Investigating the Effect of Block Length on the Performance of Fractal Coding Using Audio Files", Periodica Polytechnica Electrical Engineering and Computer Science, 64(3), pp. 303–312, 2020. https://doi.org/10.3311/PPee.14896

files and adopting techniques to reduce the encoding time and compressed audio quality [9, 10].

The contribution of this paper is to discuss the application of FC on different audio file types and the effect of block length on the audio quality and compression performance. The cost of the encoding time of the FC is out of the scope of this paper. The rest of the paper is organized as follows: Section 2 explains the basic concepts of the Fractal Coding. Section 3 shows the related works. Section 4 consists the experimental result and discussion and finally conclusion and future works are given in Section 5.

2 Basic concepts

This section explains briefly some concepts regarding Fractal Coding technique. Brief history of Fractal Coding, fractal encoding and decoding, Fractal Parameters and RMS Metric, and Fractal Parameter bits allocation are highlighted.

2.1 Fractal Coding

Fractal was discovered by Benoit Mandelbrot in 1977 [11]. Fractal geometry is the science concerning the property of fractal objects found in the real world. The fractal concept is based on similarities and redundancies in most real-world objects. In 1988, Fractal Coding was first used for image compression by Barnsley and Sloan [12] at the Georgia Institute of Technology. In the same year, Jacquin extended Barnsley and Sloan's work using the mathematics of IFSs. Jacquin finally established a practical Fractal Coding algorithm using Partition Iteration Function Systems (PIFS) [13]. The basic idea of Fractal Coding is to generate two types of blocks, namely, domain and range blocks. Each range block is matched with all the domain blocks to find the most similar one. Fractal error, Root Mean Square (RMS) is used in the matching process to find the best domain block for each range block with minimum error. Each range block is matched with the all domain blocks in order to find the most similar domain block by computing the RMS between the range and domain blocks, minimum RMS means more similarity. At the end, each range block is encoded by four parameters called Fractal Parameters (FPs). Moreover, the bit allocation for each parameter of FPs and the development of the Fractal Coding is discussed at the next Subsections 2.2, 2.3, 2.4 [14].

Fig. 1 and Fig. 2 are two examples of the Fractal Coding. The first example is the generating of Sierpinski's Triangle using FPs with 6 iterations and the second example is fractal encoding from a domain block map to a range block on a Lena image [15].

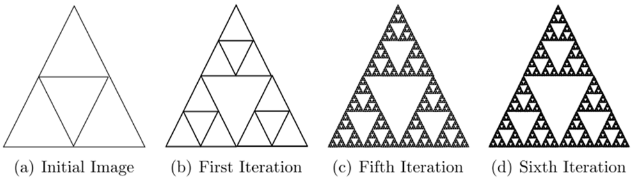

Fig. 1 The generating of Sierpinski's Triangle [15]

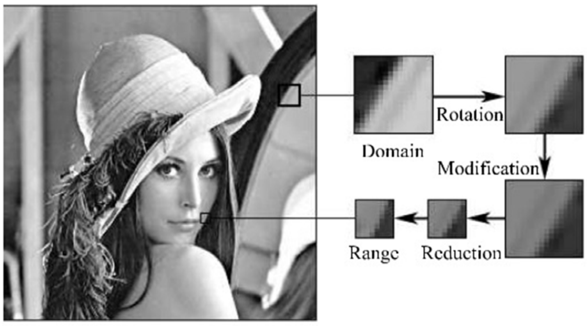

Fig. 2 Fractal encoding [15]

Fractal Coding is a prominent approach used for data compression because of its high compression ratio and the accepted fidelity of the reconstructed signal compared with that of other techniques, such as DWT and DCT [7, 16]. The main concept of Fractal Coding is similar to Vector Quantization (VQ) since both are classified under lossy, block-based, and non-transform compression techniques. However, Fractal Coding is superior to VQ in that a codebook or dictionary on the sender and receiver sides is no longer necessary. The input uncompressed audio itself is used as the codebook and considered as a virtual codebook instead of using a separate codebook such as VQ [11].

Fractal Coding also requires less computational complexity because its process does not require any transformation, unlike DCT and DWT. Moreover, Fractal Coding presents an asymmetric property in which the encoding process is time-consuming during the range–domain matching process while the decoding is simple and fast [7, 16, 17]. The long encoding time is considered the main drawback of Fractal Coding and limits its wide application as a standard compression technique. Since the fractal encoding process requires huge computation during encoding, much effort is directed to enhancing the performance of Fractal Coding in the image domain in comparison to the audio domain [9, 10, 18, 19].

2.2 Fractal encoding and decoding

A fractal object can be generated from a recursive process of different contractive mapping functions applied to a part of the object. Let W_1, W_2, W_3, ... W_N be different mapping functions and *N* is the total number of functions. General FPs can be defined as in Eq. (1):

$$A = W_1(A) \cup W_2(A) \cup W_3(A) \cup \ldots \cup W_N(A), \tag{1}$$

where *A* is the fractal object or FPs, one (*A*) for each block and each block consists of number of samples.

The main concept of Fractal Coding is that, after generating the range and domain blocks from the object, the range block can be similar to the domain block after applying a specific mapping function. The result is a set FPs; that is, one FPs for each range block [20]. As is the case in other compression techniques, Fractal Coding consists of two main processes, namely, encoding and decoding.

The encoding process of Fractal Coding includes three steps as follows:

1. Partition the input object into the non-overlapped range blocks and the same input object into overlapped blocks after making down sample operation.
2. Match between each range block with all domain blocks to produce optimum FPs for each range block with minimum RMS error based on two factors scale and offset, Eqs. (2)–(5).
3. Store the obtained set of FPs which contain the Fractal Parameters for each range block in the output file.

The decoding process is simple and straightforward and includes applying the FPs on a random initial random signal that has the same size as the original signal. This signal is used in this process to generate the range and domain blocks. Applying the FPs on the initial random signal using Eq. (6) is performed and a new audio signal is obtained and the process is repeated on the generated new audio signal for several iterations until obtaining an audio signal which is close to the original audio signal [17, 21].

2.3 Fractal Parameters and RMS metric

The basic theory of FC is that each range block can be approximately reconstructed using Fractal Parameters and Eq. (6) [22, 23].

For more specific, there are four Fractal Parameters that represent each range block which are index of domain block, scale, shift (offset) and the affine transform. Scale represents the difference between the range blocks divided by the difference between the domain blocks. Shift indicates the difference between the average of the range blocks and the average of the domain blocks. Equations (2) and (3) represent the computation of the scale and shift [19, 22-24]:

$$Scl_j = \frac{n\sum_{i=1}^{n} d_i r_i \sum_{i=1}^{n} d_i \sum_{i=1}^{n} r_i}{\sum_{i=1}^{n} d_i^2 - \left(\sum_{i=1}^{n} d_i\right)^2} \tag{2}$$

$$O_j = \frac{1}{n}\left(\sum_{i=1}^{n} r_i - Scl_j \sum_{i=1}^{n} d_i\right), \tag{3}$$

where Scl_j and O_j are the scale and the offset of the j^{th} range block; ($0 < j < N$); *N* is the number of range blocks; *n* is number of samples in the block; d_i, r_i are the value of the i^{th} sample in the domain and range block, respectively.

The affine transform is applied to the domain blocks before computing the scale, offset and RMS in order to guarantee the maximum similarity that can be achieved between the range and domain blocks. The affine mapping involves techniques such as mirroring, reflection, and rotation that can be mathematically represented in Eq. (4):

$$D_i' = T \times D_i, \tag{4}$$

where D_i' is the i^{th} transformed domain block, D_i is the i^{th} original domain block, ($0 < i < N$) *N* is number of domain blocks and *T* is the affine transform matrix as shown in Fig. 3.

In fractal image coding, there are 8 affine transforms: no change, rotation through +90, rotation through +180, rotation through −90, reflection about mid-vertical axis, reflection about mid-horizontal axis, reflection about first diagonal, reflection about second diagonal, while in Fractal Audio Coding (FAC) the number of affine transforms is limited to two affine transforms: 0, when no change and 1 when rotation through +180 as the domain block is a vector (one dimension matrix).

For instance, in the case of Fractal Audio Coding, suppose the block length is 4 samples and the samples value

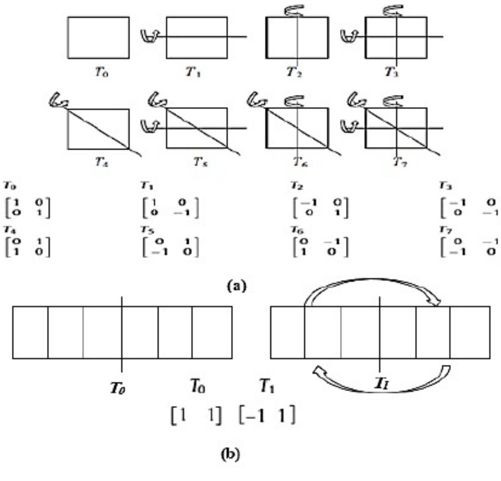

Fig. 3 Affine transforms (a) image domain, (b) audio domain

are {120, 132, 140, 90}. When the affine transform is 0, the block samples value is still same; no change occurs and the block samples value become {90, 140, 132, 120} when the affine transform is 1.

In order to find the most similar domain block for each range block with less distortion, fractal Root Mean Square (RMS) error metric is used. RMS is calculated for each range block and domain blocks using Eq. (5) and scale and offset, Eqs. (2) and (3), respectively [2, 23]:

$$\mathrm{RMS}_j^2 = \frac{1}{n}\left[\sum_{i=1}^{n} r_i^2 + Scl_j\left(Scl_j\sum_{i=1}^{n} d_i^2 - 2\sum_{i=1}^{n} d_i r_i + 2O_j\sum_{i=1}^{n} d_i\right) + O_j\left(nO_j - 2\sum_{i=1}^{n} r_i\right)\right], \tag{5}$$

where RMS_j^2 is the error between the jth range and domain block, $\{0 < j < N\}$; N is the number of domain blocks. RMS_j^2 is required to compute for each range block and all domain blocks.

The reconstruction of the approximation range block in the decoding process is represented by the following Eq. (6) which depends on the scale and shift parameters after making the appropriate affine transform on the appropriate affine transform on the d_i:

$$r_k' \approx Scl_j \times d_i + O_j, \tag{6}$$

where r_k' is the kth reconstructed range block; Scl_j and O_j are the scale and the offset parameter of the jth range block, respectively; d_i is the value of the ith sample of the arbitrary domain block, which is a block with samples has any initial values such as 0.

Later, the shift coefficient is replaced by the range mean value. The reason behind the replacement is discussed theoretically by [22, 23] since the shift value may be a positive or negative value so it needs one additional bit to represent the sign while the mean range value is a positive one. Moreover, using mean range value with scale allows the search for the best matching domain block to speed up. So, in the modified fractal technique, the parameters are scale, range mean, domain index, and the affine transform. The scale and RMS are modified and calculated by the following Eq. (7) and Eq. (10) [23, 25, 26]:

$$Scl_j = \frac{\frac{1}{n}\sum_{i=0}^{n-1} d_i r_i - \bar{d}\bar{r}}{\sigma_d^2} \tag{7}$$

$$\bar{r} = \frac{1}{n}\sum_{i=0}^{n-1} r_i, \quad \bar{d} = \frac{1}{n}\sum_{i=0}^{n-1} d_i \tag{8}$$

$$\sigma_r^2 = \frac{1}{n}\sum_{i=0}^{n-1} r_i^2 - \bar{r}^2, \quad \sigma_d^2 = \frac{1}{n}\sum_{i=0}^{n-1} d_i^2 - \bar{d}^2 \tag{9}$$

$$\mathrm{RMS}_j^2 = \sigma_r^2 + Scl_j\left[Scl_j\sigma_d^2 + 2\bar{d}\bar{r} - \frac{2}{n}\sum_{n=0}^{n-1} d_i r_i\right], \tag{10}$$

where RMS_j^2 is the error between the current range block and domain block; Scl_j is the scale of the jth range block; $\{0 < j < N\}$; N is the number of range blocks; d, r are domain and range block with n samples respectively; $d_i r_i$ is the value of the ith sample in the domain and range block, respectively; $\bar{d}, \bar{r}$ are the mean of the domain and range block, respectively; σ_d^2, σ_r^2 are the variances of the domain and range block, respectively σ_d^2, σ_r^2.

The reconstructed range block is obtained by applying the appropriate affine transform on d_i then using the following decoding equation, Eq. (11):

$$r_k' = Scl_j\left(d_i - \bar{d}\right) + \bar{r}, \tag{11}$$

where r_k' is the kth retrieved range block; Scl_j is the scale parameter of the jth range block; d_i is the value of the ith sample of the arbitrary domain block which is a block with samples has any initial values such as 0; $\bar{d}, \bar{r}$ are the mean of the arbitrary domain and range blocks, respectively. The proposed model in this study adopts the modified Fractal Coding proposed by [22, 25].

2.4 Fractal Parameter bits allocation

The main objective of the Fractal Coding is encoding or compression. In order to know how the Fractal Coding can compress the data, it is necessary to elaborate the bit allocation for each FPs.

When the fractal encoding process finishes, the output is the set of FPs, one FPs for each range block which is stored in a file that represents the compressed file of the original signal. The number of range blocks depends on the block length. Each FPs consists of:

1. scale value,
2. range mean value,
3. index of domain block,
4. affine transform.

Each of the FPs is represented by a different number of bits depending on the index of the domain blocks which is, in total, less than the number of bits required to represent the actual range block. The number of the allocated bits for each parameter is as follows:

1. Scale value – 6 bits.
2. Mean of the range block – 8 bits.

3. Domain block index – the number of bits allocated for this factor depends on the total number of audio file samples.
4. Affine transform – 1 bit.

3 Related works

This section is dedicated to showing the related works that discussed the application of Fractal Coding on the audio files.

The Fractal Audio Coding (FAC) was first discussed by Van de Walle [27] in 1995. Then, in 1997, Wannamaker and Vrscay [28] utilized a Wavelet Transform with the FC to obtain audio compression. They used the hybrid method between FC and WT since the audio signals have more smoothness than images. Their approach achieved a 6:1 compression ratio with good signal reconstruction fidelity.

Xiao [8] presented the implementation and discussion of the Fractal Coding on an audio signal in his master's thesis. He carried out empirical experiments to test the effect of this technique on audio signals instead of images. This study concluded that a fractal model is not appropriate to apply alone on audio signals and requires some improvement of a technique for successful application. In addition, the performance is controlled by several parameters such as domain creation, range partitioning, bit allocation and others. The achieved compression factor in this study is around 3 to 6.

George and Salih [29] proposed an audio compression system using FC based on PIFS and affine transformation using a PCM wave file. They showed the relation of block length and jumping step with encoding time, PSNR, MSE and compression ratio. In addition, they highlighted that the IFS coefficients (scale and offset) have a great effect on the compression ratio when using quantization but do not have on encoding time.

Ali et al. [9] proposed an approach to speeding up the FAC. They adopted a technique whereby the audio signal is divided into sub-signals and applies FC separately utilizing a parallel algorithm. In addition, they used a stopping condition to stop the matching process depending on specific thresholds. The proposed approach reduced the encoding time by 70 % and they conclude that the encoding time has a relationship with block length and stopping condition.

Bedan and George [10] presented a FAC approach by adopting a filtering method using moment descriptors to minimize the domain pool and speed up the matching process. The approach uses first and third order moment for classification. The results presented in this research are considered to offer a high reduction in the encoding time obtained for the PIFS without significant degradation to the audio quality.

In comparison, between fractal image and audio coding, there have been many articles published on fractal image coding [18, 19, 21, 30–32] while there is less on Fractal Audio Coding as presented above. Further research is required in the area of Fractal Audio Coding in terms of encoding time, compressed audio quality and compression ratio in order to effectively use with the audio files and stand side by side with other audio compression techniques.

4 FC model

FC model consists of two modules: encoding and decoding. These modules are shown in Fig. 4. Subsection 4.1 describes the details of these modules.

4.1 Encoding module

The first module in the model of the FAC is the encoding module. This module begins by selecting an audio signal as input for the partitioning process. In preprocessing, the range and domain blocks are constructed and the mean and variance of each range and domain block are computed. Then the matching process is executed to find the most similar blocks among the domain blocks to the particular range block.

The selecting process is executed after finishing the matching process which is used to save the information about the most similar domain block. After finishing

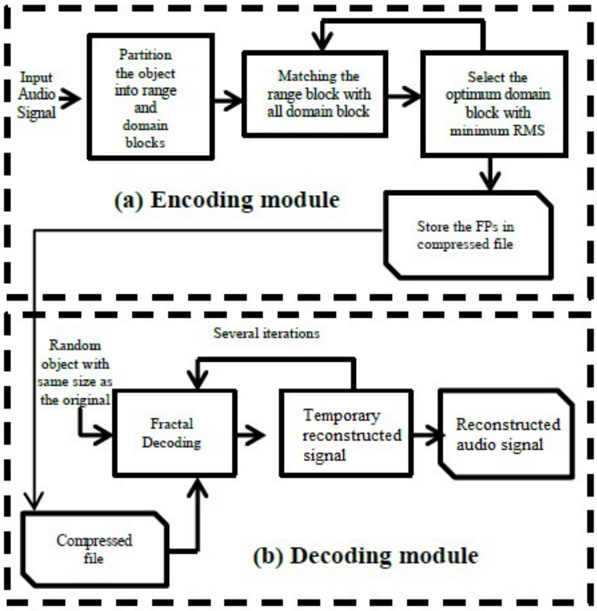
Input Audio Signal
Partition the object into range and domain blocks
Matching the range block with all domain block
Select the optimum domain block with minimum RMS
(a) Encoding module
Store the FPs in compressed file
Several iterations
Random object with same size as the original
Fractal Decoding
Temporary reconstructed signal
Reconstructed audio signal
Compressed file
(b) Decoding module

Fig. 4 FC model, (a) encoding module and (b) decoding module

the matching and selecting process, the FPs for each range block is generated and then stored into a file which represents the compressed audio signal. The flowchart of this module is shown in Fig. 4 (a).

4.2 Decoding module

The second module in FC model is the decoding module. It begins with opening the compressed audio signal and reading the set of FPs. Then, the initial audio signal that has the same size as the original audio signal is generated. After that, the set of FPs is applied to the initial random signal and get a new audio signal. The final step is to repeat this process for several iterations until obtaining an audio signal which is approximately similar to the original audio signal and saving it in an audio file.

5 Experimental results and discussion

This section is dedicated to show the result of applying Fractal Coding on different audio files and the relationship between the block length and compression performance and audio quality. This section includes the experimental setup, evaluation criteria that are used in the experiments, the dataset, and the conducted experiments.

The prototype of the Fractal Coding is developed using Java Eclipse EE IDE on the Intel® Core™ i5-4590 CPU @ 3.30 GHz 4GB RAM. MATLAB R2013a and Adobe Audition 1.5 software are also used to specify the duration of the audio files required for each experiment. Mean Square Error (MSE) [33] and Peak Signal to Noise Ratio (PSNR) [33] are used to evaluate the compressed audio quality while compression ratio and factor [34] are used to evaluate the compression performance. Regarding the audio dataset, a GTZAN dataset [35] is adopted. This dataset is used in evaluation since it is composed of two file types: speech, and music. The specifications of the audio files used in the experiments are listed in Table 1.

There are two experiments in this section. The first experiment aims to investigate the quality of the compressed audio files using Fractal Coding. The second is to present the compression ratio and factor that can be achieved using this technique with the audio signal.

Table 1 Audio files specification

Specification	
Bit per sample	16
Sample rate	44100
Channel	Mono
Audio type	Speech Music
Duration in Seconds	1–10

5.1 Audio quality

This experiment is to evaluate the quality of the audio files. MSE and PSNR are used in this experiment. Two types of audio files are selected which are speech and music. Six audio files are chosen, three music and three speech files, with a 44100 sample rate, 16 bits per sample and one second period of time as shown in Table 2.

Any block length can be used in this experiment. Thus, block length of 10, 20, 30, 40, 50 are selected. As shown from Table 2, the quality of the audio file

Table 2 Effect of block length on the quality of the compressed files using Fractal Coding

Original File name	Original file size (bits)	Compressed file size (bits)	Block length sample	Kbit/s	MSE	PSNR
Female	705600	132347	10	132.3	1.8	45.4
Male					2.2	44.6
Voice					2.8	43.5
Jazz					3.16	43.1
Vlobos					1.97	45.1
Kalimba					3.06	43.2
Female		66197	20	66.1	3.5	42.6
Male					9.7	38.2
Voice					11.2	37.6
Jazz					6.38	40
Vlobos					5.05	41
Kalimba					10.6	37.8
Female		44147	30	44.1	14.7	36.4
Male					11.6	37.4
Voice					22.0	34.6
Jazz					6.82	39.7
Vlobos					51.2	31
Kalimba					24.7	34.2
Female		33107	40	33.1	31.5	33.1
Male					40.9	32
Voice					39	32.2
Jazz					17.9	35.5
Vlobos					78.1	29.1
Kalimba					39.3	32.1
Female		26507	50	26.5	29.5	33.4
Male					56.2	30.6
Voice					40	32
Jazz					13	36.9
Vlobos					77.6	29
Kalimba					39	32.2

has a reverse relationship with the block length; when the block length increases, the quality of the audio signal decreases and vice versa. The presentation of the relationship between the block length and the audio quality is depicted in Fig. 5.

The results in Table 2 and Fig. 5 show that Fractal Coding can be used as a compression technique to compress the audio files with high quality, PSNR 39 dB, on average when the block length is 20 samples and acceptable quality of PSNR 35 dB with block length is 30 samples.

On the other hand, in order to visually present the effect of the block length on the quality of the compressed audio signal, Fig. 6 and Fig. 7 show the comparison between the original and the compressed audio signal for two different audio files, female and Voice using different block lengths. In these two figures, the signal (a) represents the original audio files while the signals (b)–(f) represent the compressed audio signals using a block length of 10, 20, 30, 40 and 50 samples, respectively.

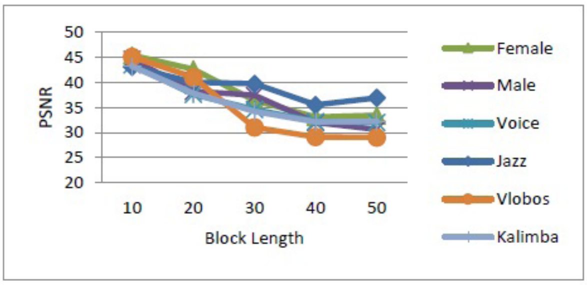

Fig. 5 Relationship between block length and audio quality PSNR

From Fig. 6 and Fig. 7, it can be observed that not much difference could be distinguished between the original signal (a) and the signals using block length of 10, 20, 30 (b)–(d).

Based on the result in Table 2, Figs. 5, 6 and 7, it can be concluded that the block length is considered the main factor that controls the quality of the compressed audio signal since it controls the quality of the compressed audio quality and the Fractal Coding can be applied directly to compress any speech or music audio files with high quality of the compressed audio files depending on the block length.

5.2 Compression performance

This subsection is to present the compression performance which are compression ratio and factor that can be achieved using Fractal Coding. Different audio signals are used which are the same files as used in part one. As in experiment Section 5.1, any block length can be used and this depends on the required compression ratio and quality of the compressed audio files. In this experiment, a different block length 10, 20, 30, 40 and 50 samples is selected to inspect the effect on the compression ratio and factor with the quality of the audio signal after decompression. Table 3 shows the results of this experiment.

Table 3 explains the proportional relationship between the block length and the compression ratio and factor. The compression ratio and factor increase when the block length increases. The reason for this relationship is because of the Fractal Coding algorithm partitions the input signal into non-overlapped range blocks and encodes them into a set of FSs.

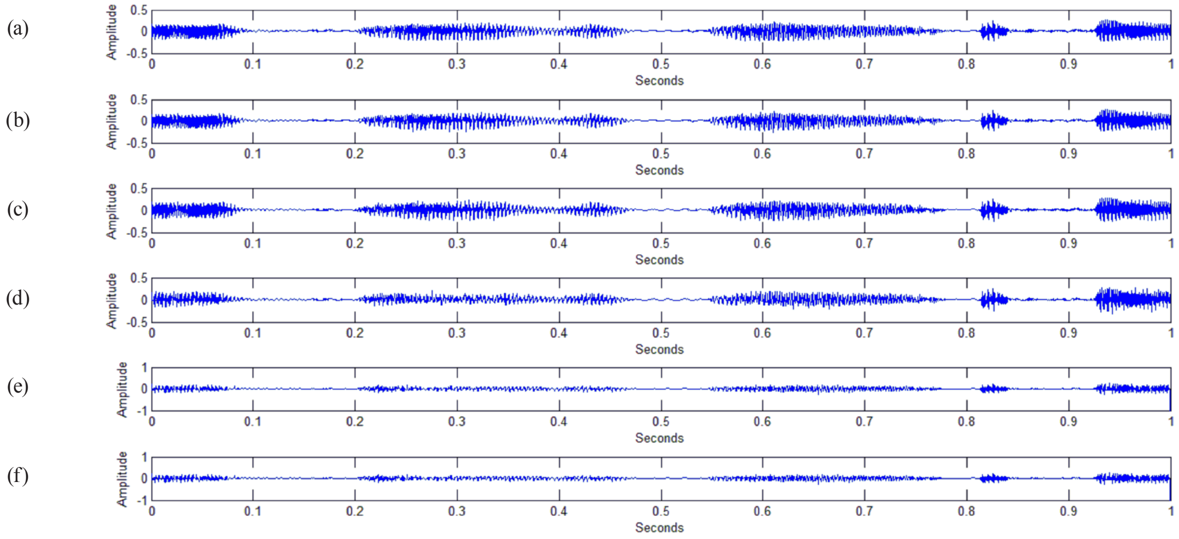

Fig. 6 Quality of the compressed audio signal using different block length and Female audio signal, (a) original file, (b), (c), (d), (e), (f) are the compressed audio file with 10, 20, 30, 40 and 50 samples block length

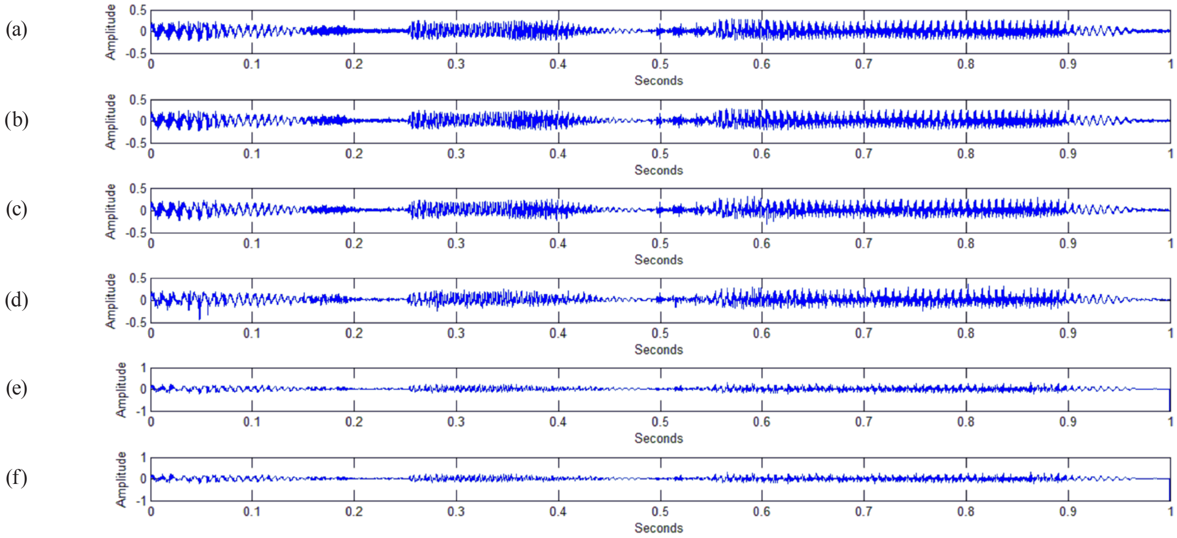

Fig. 7 Quality of the compressed audio signal using different block length and Voice audio signal, (a) original file, (b), (c), (d), (e), (f) are the compressed audio file with 10, 20, 30, 40 and 50 samples block length

Table 3 Effect of the block length on the compression ratio using Fractal Coding

Original File name	Original file size (bits)	Compressed file size (bits)	Block length sample	Compression Ratio %	Compression Factor
Female Jazz Voice	705600	132347	10	81.2	5.33
Female Jazz Voice		66197	20	90.62	10.66
Female Jazz Voice		44147	30	93.7	15.98
Female Jazz Voice		33107	40	95.3	21.31
Female Jazz Voice		26507	50	96.2	26.61

Thus, when the block length increases, the number of range block decreases and the set of FPs that represents these range blocks and the number of bits required representing them will decrease. This means decreasing in the compressed file size and at the same time increasing in the compression ratio.

Figs. 8 and 9 highlight the relationship between the block length and compression ratio and factor.

It is clear from Figs. 8 and 9 that increasing the block length leds to increasing the compression ratio and factor. When the block length is 50 samples, the achieved compression ratio is about 96 % and compression factor is about 26.

Based on the experiments in Subsections 4.1 and 4.2, the achieved compression ratio using the Fractal Coding

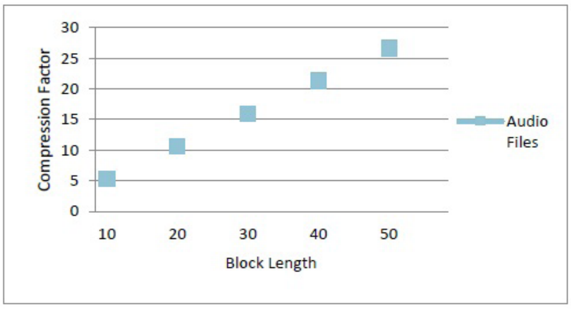

Fig. 8 Relationship between block length and audio quality PSNR

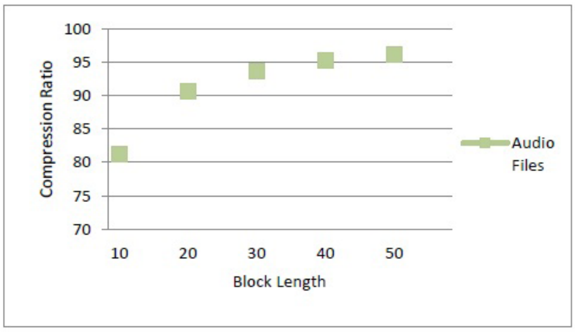

Fig. 9 Relationship between block length and audio quality PSNR

algorithm is around 90 % and compression factor is around 10 since the original audio size, in this case, is 705600 bits, the compressed size 66197 bits as shown in Table 3. So the difference in size is 639403 bits which represent the compression ratio of around 90.62 % of the original audio size. The quality of the compressed audio files PSNR, in this case, is considered acceptable of around 39 dB using 20 samples block length, as shown in Table 2

6 Conclusion

Fractal Coding is considered as a prominent technique to compress the audio files since it can achieve high compression ratio and acceptable audio quality. This paper is dedicated to explore the relationship between the block length with the audio quality and compression performance. The results exhibit that there is inverse relationship between block length and audio quality and proportional relationship between block length and compression ratio and factor. Furthermore, it can be noticed that the Fractal Coding can be compressed any speech and music audio signal directly with acceptable quality, PSNR 39 dB on average with high compression ratio around 90 % with compression factor a round 10 when the block length is 20 samples. In order to increase the compression ratio, it needs to increase the block length and this leds to decrease the audio quality. In future work, new method should be adopted to maintain the audio quality along with the high compression ratio when increasing the block length. Moreover, the relationship between the block length and encoding time will be investigated.

Acknowledgment

This research is supported by the Ministry of Higher Education and Scientific Research \ Studies Planning and Follow-up Directorate and Middle Technical University \ Electrical Engineering Technical College, Iraq.

References

[1] Tang, M., Zeng, S., Chen, X., Hu, J., Du, Y. "An adaptive image steganography using AMBTC compression and interpolation technique", Optik, 127(1), pp. 471–477, 2016. https://doi.org/10.1016/j.ijleo.2015.09.216

[2] Ibaida, A., Al-Shammary, D., Khalil, I. "Cloud enabled fractal based ECG compression in wireless body sensor networks", Future Generation Computer Systems, 35, pp. 91–101, 2014. https://doi.org/10.1016/j.future.2013.12.025

[3] Sayood, K. "Introduction to Data Compression", The Morgan Kaufmann Series in Multimedia Information and Systems, Elsevier, Waltham, MA, USA, 2012. https://doi.org/10.1016/C2010-0-69630-1

[4] Kodituwakku, S. R., Amarasinghe, U. S. "Comparison of Lossless Data Compression Algorithms for Text Data", Indian Journal of Computer Science and Engineering, 1(4), pp. 416–425, 2010.

[5] Chang, C.-C., Nguyen, T.-S., Lin, M.-C., Lin, C.-C. "A novel data-hiding and compression scheme based on block classification of SMVQ indices", Digital Signal Processing, 51, pp. 142–155, 2016. https://doi.org/10.1016/j.dsp.2015.11.007

[6] Lin, C.-C., Liu, X.-L, Tai, W.-L., Yuan, S.-M. "A novel reversible data hiding scheme based on AMBTC compression technique", Multimedia Tools and Applications, 74(11), pp. 3823–3842, 2015. https://doi.org/10.1007/s11042-013-1801-5

[7] Sheltami, T., Musaddiq, M., Shakshuki, E. "Data compression techniques in Wireless Sensor Networks", Future Generation Computer Systems, 64, pp. 151–162, 2016. https://doi.org/10.1016/j.future.2016.01.015

[8] Xiao, H. "Fractal Audio Coding", Master's Thesis, Queen's University, Canada, 2005.

[9] Ali, S. M., George, L. E., Taher, H. B. "Speeding up Audio Fractal Compression", International Journal of Advanced Research in Computer Science and Software Engineering, 3(6), pp. 86–92, 2013.

[10] Bedan, A. K., George, L. E. "Speeding-up Fractal Audio Compression Using Moment Descriptors", International Journal of Scientific & Engineering Research, 4(7), pp. 1623–1627, 2013.

[11] Nelson, M., Gailly, J.-L. "The Data Compression Book", M&T Books, New York, NY, USA, 1995.

[12] Barnsley, M. F., Sloan, A. D. "A Better Way to Compress Images", Managing Megabytes, 13, pp. 215–223, 1988.

[13] Jacquin, A. E. "Image coding based on a fractal theory of iterated contractive image transformations", IEEE Transactions on Image Processing, 1(1), pp. 18–30, 1992. https://doi.org/10.1109/83.128028

[14] Ali, A. H., Abbas, A. N., George, L. E., Mokhtar, M. R. "Image and Audio Fractal Compression: Comprehensive Review, Enhancements and Research Directions", Indonesian Journal of Electrical Engineering and Computer Science, 15(3), pp. 1564–1570, 2019. https://doi.org/10.11591/ijeecs.v15.i3.pp1564-1570

[15] Fisher, Y. "Fractal Image Compression", Fractals, 2(3), pp. 347–361, 1994. https://doi.org/10.1142/S0218348X94000442

[16] Mammeri, A., Hadjou, B., Khoumsi, A. "A Survey of Image Compression Algorithms for Visual Sensor Networks", International Scholarly Research Notices, 2012, Article ID: 760320, 2012. https://doi.org/10.5402/2012/760320

[17] Wang, J., Zheng, N. "A Novel Fractal Image Compression Scheme With Block Classification and Sorting Based on Pearson's Correlation Coefficient", IEEE Transactions on Image Processing, 22(9), pp. 3690–3702, 2013. https://doi.org/10.1109/TIP.2013.2268977

[18] Wang, X., Zhang, D., Guo, X. "Novel hybrid fractal image encoding algorithm using standard deviation and DCT coefficients", Nonlinear Dynamics, 73(1–2), pp. 347–355, 2013. https://doi.org/10.1007/s11071-013-0790-2

[19] Jaferzadeh, K., Moon, I., Gholami, S. "Enhancing fractal image compression speed using local features for reducing search space", Pattern Analysis and Applications, 20(4), pp. 1119–1128, 2017. https://doi.org/10.1007/s10044-016-0551-1

[20] Johri, P., Kumar, A., Amba "Review paper on text and audio steganography using GA", In: International Conference on Computing, Communication & Automation, Noida, India, 2015, pp. 190–192. https://doi.org/10.1109/CCAA.2015.7148403

[21] Kamble, S. D., Thakur, N. V., Malik, L. G., Bajaj, P. R. "Color Video Compression Based on Fractal Coding Using Quadtree Weighted Finite Automata", In: Mandal, J. K., Satapathy, S. C., Kumar Sanyal, M., Sarkar, P. P., Mukhopadhyay, A. (eds.) Information Systems Design and Intelligent Applications, Springer, New Delhi, India, 2015, pp. 649–658. https://doi.org/10.1007/978-81-322-2247-7_66

[22] Tong, C. S., Pi, M. "Fast fractal image encoding based on adaptive search", IEEE Transactions on Image Processing, 10(9), pp. 1269–1277, 2001. https://doi.org/10.1109/83.941851

[23] George, L. E. "IFS Coding for Zero-Mean Image Blocks", Iraqi Journal of Science, 47(1), pp. 190–194, 2006.

[24] Koli, N. A., Ali, M. S. "A Survey on Fractal Image Compression Key Issues", Information Technology Journal, 7(8), pp. 1085–1095, 2008. https://doi.org/10.3923/itj.2008.1085.1095

[25] Al-Hilo, E. A., George, L. E. "Speeding-up Fractal Colored Image Compression Using Moments Features", In: 2008 Digital Image Computing: Techniques and Applications, Canberra, ACT, Australia, 2008, pp. 486–490. https://doi.org/10.1109/DICTA.2008.18

[26] Rivas, E. "Fourier phase domain steganography: Phase bin encoding via interpolation", In: Defense and Security Symposium, Orlando, FL, USA, 2007, Article number: 65790W. https://doi.org/10.1117/12.719512

[27] Van de Walle, A. "Relating Fractal Image Compression to Transform Methods", PhD Thesis, University of Waterloo, 1995.

[28] Wannamaker, R. A., Vrscay, E. R. "Fractal Wavelet Compression of Audio Signals", Journal of the Audio Engineering Society, 45(7–8), pp. 540–553, 1997.

[29] George, L. E., Salih, H. M. "Audio Compression Using Fractal Coding", Diyala Journal For Pure Sciences, 6(1), pp. 1–10, 2010.

[30] Wu, M.-S. "Genetic algorithm based on discrete wavelet transformation for fractal image compression", Journal of Visual Communication and Image Representation, 25(8), pp. 1835–1841, 2014. https://doi.org/10.1016/j.jvcir.2014.09.001

[31] Das, S., Ghoshal, D. "A Proposed Hybrid Color Image Compression Based on Fractal Coding with Quadtree and Discrete Cosine Transform", International Journal of Computer, Electrical, Automation, Control and Information Engineering, 9(11), pp. 2366–2373, 2015. https://doi.org/10.5281/zenodo.1110923

[32] Valarmathi, M. L., Sobia, M., Devi, R. B. "Iteration-Free Fractal Image Compression Using Pearson's Correlation Coefficient-Based Classification", In: Rajsingh, E. B., Bhojan, A., Peter, J. D. (eds.) Informatics and Communication Technologies for Societal Development, Springer, New Delhi, India, 2015, pp. 157–166. https://doi.org/10.1007/978-81-322-1916-3_16

[33] Ma, X., Pan, Z., Hu, S., Wang, L. "Reversible data hiding scheme for VQ indices based on modified locally adaptive coding and double-layer embedding strategy", Journal of Visual Communication and Image Representation, 28, pp. 60–70, 2015. https://doi.org/10.1016/j.jvcir.2015.01.009

[34] Salomon, D., Motta, G. "Handbook of Data Compression", Springer, London, UK, 2010. https://doi.org/10.1007/978-1-84882-903-9

[35] Tzanetakis, G. "MARSYAS: Music Analysis, Retrieval and Synthesis for Audio Signals", [online] Available at: http://marsyas.info/downloads/datasets.html [Accessed: 23 October 2019]