Compression and Encryption Algorithms for Image Satellite Communication

Dr. Emad S. Othman, Dr. Mohammed M. Sakre

Abstract— Satellite communication systems are now designed according to an outdated Shannon information theory where all data is transmitted in meaningless bit streams. The advantages for satellite communication may include: high image compression and unbreakable encryption. So, this paper combines the compression and encryption techniques into two phases, encoding and decoding phase. The fractal compression technique is selected to compress the aerial images due its high compression capability and the Enhanced Hill Multimedia Cryptosystem (EHMC) is used for encryption to perform the goal. The fractal image compression of the input aerial image is the first stage in the encoding phase. The fractal compression technique divides the original image into two different block sizes called range and domain blocks. The best match between the range and domain blocks is known as the transformation mapping. All the transformation mappings are recorded as the output compressed file of the input aerial image.

The encryption which was done by the EHMC is the second encoding stage The EHMC belongs to the private-key techniques and works with linear transformation theory is applied on the output-compressed file. At the decoding phase, the secured compressed image is decrypted by the EHMC. Finally, the original image is reconstructed from the highly secured compressed file using the fractal decompression process. The objective of the paper is to provide an approach in the area of robust encrypted compression techniques, which achieve the privacy and confidentiality of aerial and remote sensing images transmitted over an insecure channel from unauthorized access.

Index Terms— Cryptosystem, fractal compression, image encoding, matrix key, satellite communication.
.

1 Introduction

FRACTAL compression is a lossy compression method for digital images, based on fractals. The method is best suited for textures and natural images, relying on the fact that parts of an image often resemble other parts of the same image. Fractal algorithms convert these parts into mathematical data called "fractal codes" which are used to recreate the encoded image. So, fractal image compression offers some interesting features which makes it an interesting candidate image compression technique for aerial images such as: resolution- independent, fast decoding and good image quality at low bit-rates.

The concept of fractal was introduced by Hutchinson [1] as an alternative to the traditional Euclidean geometry mainly dealing with shapes generated by nature. In the recent years, the interest of applying this theory has been steadily growing. Iterated Function System (IFS) has been used to generate and describe man-made fractal-like structures and natural images. Barnsley et. al. [2] were the first to present the concept of fractal image compression using IFS. A fully automatic image compression algorithm for real-world gray scale images called fractal block coding (FBC) was proposed by Jacquin [3]. Many important research results on this topic are collected in fisher's book [4]. Sayood [5] indicated that the main point of FBC is that it can capture and exploit a special kind of image redundancy – piecewise self-similarity in images, which is not used by traditional image coding techniques. In general, the natural images are not exactly self-similar thus some transformations are needed to reconstruct the image from transformed 'parts' of itself. The fractal code of the image is actually the collection of these transformations, which are called fractal transformations. Since fewer bits of the fractal transformations can represent the original image, high compression ratios can be achieved.

On the other hand, the fractal compressed image is encrypted by EHMC as the second encoding stage of the proposed scheme. The EHMC belongs to the private-key techniques and works with linear transformation theory. The EHMC algorithm is applied on the fractal transformations to achieve the required goal: secured compressed Aerial images.

The whole operation is done through two main phases, encoding and decoding phase. Each phase has two stages; the first encoding stage compresses the input aerial image using the fractal algorithm. The second encoding stage encrypts the compressed aerial image by EHMC algorithm before transmitting it through the media channel.

At the decoding phase, the secured compressed image is decrypted by the EHMC. Upon receiving the encrypted compressed aerial image, the EHMC decryption algorithm process takes place as the first decoding stage. The decompression fractal process takes place as the last stage in the decoding phase to reconstruct the image. The major stages are shown in Fig. 1.

- Dr. Emad S. Othman, Senior Member IEEE - Region 8, High Institute for Computers and Information Systems, AL-Shorouk Academy, Cairo – Egypt, PH- 0020-01121024270. E-mail: emad91@hotmail.com
- Associate Professor. Mohammed M. Sakre, Senior Member IEEE - Region 8, High Institute for Computers and Information Systems, AL-Shorouk Academy, Cairo – Egypt, PH- 0020-01006525776. E-mail: m_sakre2001@yahoo.com

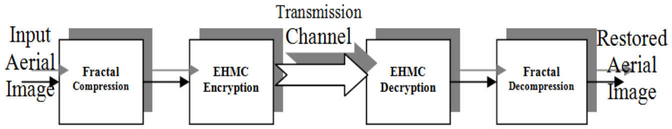


Fig. 1 The flow of image encoding/decoding phases

The following section describes the encoding phase process of the proposed scheme including the compression and encryption algorithms. . Section 3 explains the decoding phase process that contains the decryption and decompression of the image and the required procedures to reconstruct the original image. Section 4 contains some results of the work and finally, the conclusions are presented in section 5.

2 Encoding Phase

The encoding phase has two major stages: image compression and encryption stages. The following subsections describe the two stages of the encoding phase of the proposed process for the input aerial image at the transmitter site.

- **Fractal Image Compression Stage**

Fractal image compression exploits similarities within images. These similarities are described by a contractive transformation *T* of the image whose fixed point is close to the image itself. A transformation *T* is called contractive if the distance *d* between two arbitrary points *X*, *Y* is becoming smaller after applying *T* to them:

$$d\,(X, Y) \geq d(T(X), T(Y)) \qquad (1)$$

The image transformation consists of block transformations, which approximate smaller parts of the image by larger ones using contractive affine transforms [6]. The smaller parts are called ranges and the larger ones are domains. Each range forms a partition of the image while the total ranges (range-pool) represent the whole image. The domains can be selected freely within the image and may overlap. Fig. 2(a) shows a non-over- lapping domains and Fig. 2(b) shows an overlapping domain-pool for an image of size 512 × 512.

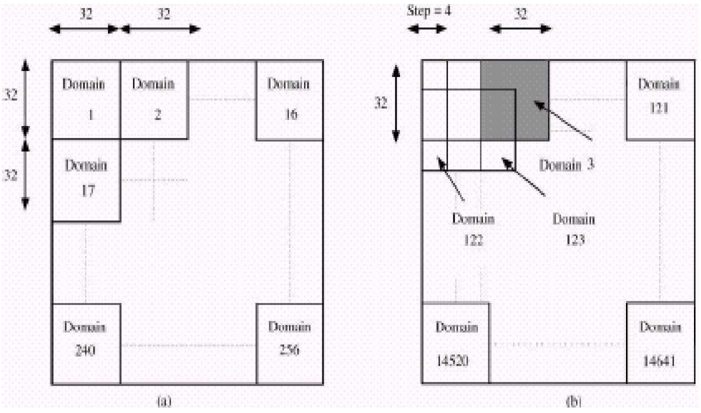


Fig. 2 Domain pools of an image of size 512×512:
(a) Non-overlapping ranges,(b) Overlapping domain-pool situated on a lattice with spacing step = 4.

An appropriate domain must be found for each range. The domain blocks are transformed to match a given range block as closely as possible [7]. Usually the root mean square error (*rms*) is used to compare the difference threshold error between range and domain blocks. The transformation applied to select the domain pool before computing the *rms* includes of the following parts:

- Geometrical contraction (usually implemented by down sampling the domain).
- Affine motion (modeled by using the 8 isometrics of the square block).
- Gray value adaptation. A least square optimization is performed in order to determine the best values for the parameters s and o describing contrast and brightness modifications, respectively. The contrast *s* and brightness *o* of a domain block are adapted to match the given range block by minimizing the rms-error between the transformed domain and the range block. The least square error optimization has to be differentiated with respect to *s* and *o*, then the resulting equations are given by:

$$s = \frac{n\sum_{i=1}^{n} a_i b_i - \sum_{i=1}^{n} a_i \sum_{i=1}^{n} bi}{n\sum_{i=1}^{n} a_i - \left(\sum_{i=1}^{n} a_i\right)^2} \qquad (2)$$

and

$$o = \frac{\sum_{i=1}^{n} b_i - s\sum_{i=1}^{n} a_i}{n} \qquad (3)$$

Where, a_i and b_i are the pixels intensities in the domain and range block respectively. These two values of *s* and *o* represent the least square solutions of the above minimization problem. The contrast adaptation *s* then the brightness shift *o* are both applied to the domain where the result matches the range block within certain threshold error [8].

Afterwards *s* and *o* are quantized and the output compressed file contains the stored data representing each range block consists of the location of the corresponding domain, the isometric being used, the values *s* and *o* for contrast and brightness. Fig.3 shows the selection of two blocks from Lenna image, (a) represents a range block while (b) represents the domain block.

The shape of ranges and domains depends on the applied partitioning algorithm being used during the compression process [9]. This paper uses the adaptive quadtree partitioning to obtain the range-domain best match.

The range-domain block error is compared to a predefined threshold called the collage error. If the calculated range-domain error located at the first quadtree level is greater than the threshold then the range block is divided into four smaller range blocks located at the second quadtree level [10]. The ranges at the second level are compared once more with the domain blocks and the difference error is calculated again. The process is repeated at different levels until the error is less than the given threshold error the best rang-domain match is found.

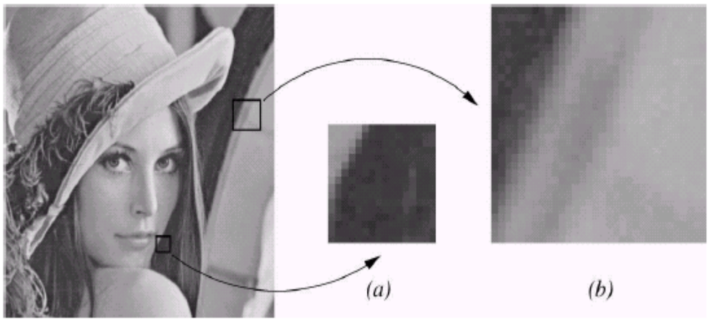


Fig. 3 Two partitions of Lenna image which are considered to be similar:
(a) Rang block can be described using domain block (b). The whole domain block (b) has to be shrunk to match the size of (a) and the luminance values (contrast and brightness) have to be inverted.

A quick view to the adaptive quadtree algorithm is as follows:

Adaptive Quadtree Algorithm

```
Begin
    Load image
    Create initial range and domain pools
    Do
        For (all ranges in the current quadtree level)
            For (all domains in the current quadtree level)
                    Compute rms(range, domain)
                    If (smallest rms ≤ collage error) OR
                    (range size is minimal)
                         Store transformation parameters
                    Else
                         Split range into its 4 quadrants (for next
                    quadtree level)

              Create range and domain-pool for the next quadtree level
        Until (all allowed quadtree levels have been computed)
        Return transformation parameters
End
```

After being compressed with the fractal algorithm using the quad tree partitioning, the compressed transformation of the original input image is encrypted by the EHMC algorithm.

- **The EHMC Encryption Stage**

This is the second encoding stage of the process. The EHMC algorithm which, belongs to block ciphers deals with the compressed fractal image using linear transformation. Once the key is selected, each character in the key is mapped to a unique character using linear algebra to form a square invertible matrix with unlimited size [11]. The Matrix coefficients are independent of the input compressed image transformation. The image partitioning block size equals the square root of the key length selected.

The compressed fractal image is segmented into *n* segments, each of width *m*, forming an input matrix of order *mxn*. The input matrix *X* is encrypted using the listed algorithm:

Taking an invertible *mxn* matrix as a key. This key is generated from a random source of integer number having the following properties :

(I) $\left| K \right| \neq 0$, where $\left| K \right|$ is the matrix determent.
(II) *K* is a singular matrix.
(III) The greatest common divisor (gcd) between the determent of the matrix *K* and 256 must equal to one. In short, *gcd (|K|, 256) = 1.*

If the width of the last segment does not equal to *m*, this segment is padded with zeros. The encrypted matrix *Y* of order *mxn* is obtained using the linear transformation as:

$$\begin{bmatrix} y_{11} & y_{12} & \cdots & y_{1n} \\ y_{21} & y_{22} & \cdots & y_{2n} \\ \cdot & \cdot & & \cdot \\ \cdot & \cdot & & \cdot \\ y_{m1} & y_{m2} & \cdots & y_{mn} \end{bmatrix} = \begin{bmatrix} k_{11} & k_{12} & \cdots & k_{1m} \\ k_{21} & k_{22} & \cdots & k_{2m} \\ \cdot & \cdot & & \cdot \\ \cdot & \cdot & & \cdot \\ k_{m1} & k_{m2} & \cdots & k_{mm} \end{bmatrix} \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1n} \\ x_{21} & x_{22} & \cdots & x_{2n} \\ \cdot & \cdot & & \cdot \\ \cdot & \cdot & & \cdot \\ x_{m1} & x_{m2} & \cdots & x_{mn} \end{bmatrix} \; Mod\ 256 \qquad (4)$$

Where $X_q = (x_{1q}, x_{2q}, \ldots, x_{mq})$, $Y_q = (y_{1q}, y_{2q}, \ldots, y_{mq})$, $q = 1, 2, \ldots, n$ and *n* is the number of segments in the compressed image. In other words, the matrix encryption algorithm can be described as *Y = K X mod 256*. Now, the secured encrypted compressed image is ready to be transmitted to the receiver through the media channel where the second phase of the process takes place as explained below.

3 Decoding Phase

The decoding phase has two major stages: decryption and decompression stages. Upon receiving the encrypted compressed image at the ground base station, the decryption algorithm is applied to it as follows:

1. The decryptor calculates the inverse matrix of the predetermined key.
2. The encrypted compressed image is divided into *n* segments each of width *m*.
3. Applying the formula $X = K^{-1}\ Y \bmod 256$ to retrieve the compressed image.

Then, the decompression process takes place after decryption. Decompressing the information in the fractal-compressed file is less complex than the compression process. Given the fractal code of an image, the image reconstruction is done by iterating the transformation on an arbitrary image. Each domain block of the initial image is transformed to its corresponding range block and the final value of the pixel range intensity z is given by :

$$z = s_i \times d_{ij} + o_i, \qquad (5)$$

where d_{ij} is the domain pixel value at *(i, j).*

As the decoding proceeds, the image converges to a stable image [12]. Usually 4 iterations are adequate to reconstruct the decoded image. To measure the produced image quality, the peak to peak signal to noise ratio (PSNR) is used. For 8 bit gray scale images the PSNR measured in dB is defined by:

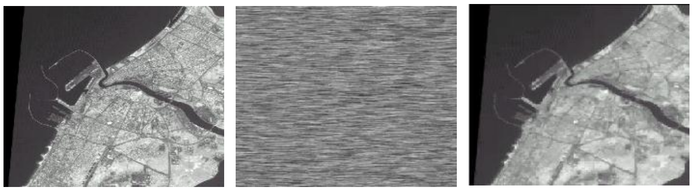
$$\mathrm{PSNR} = 10\log\frac{255^2}{rms - error} = 10\log\frac{255^2}{\frac{1}{\#\,pixels}\sum(P'_{ij} - P_{ij})^2} \qquad (6)$$

Where p_{ij} and p'_{ij} denote the pixel intensities in the original and the reconstructed images respectively.

4 Experiments and Results

Numerous experiments were done in the laboratory using aerial images and other standard satellite images. In developing test scenarios for the presented model, it is found that the fractal compression performs very efficiently with aerial images compared to other standard images due to the high data redundancy exist in the aerial images. The EHMC as an encryption algorithm, which belongs to the private-key techniques and works with linear transformation theory is combined with the fractal scheme to fulfill the required task.

By applying the scheme on Dubai City 1024 × 1024, the results are shown in Fig. 4. Fig.4 (a) the input Original aerial image 1024 × 1024 for Dubai City; (b) the compressed encrypted image output from the first stage of the scheme; (c) the restored aerial image proceed by the second scheme with CR = 107 and PNSR = 21.55 dB.



(a) Original image (b) Encrypted (c) Restored

Fig. 4 (a) Original aerial image 1024 x 1024 for Dubai City; (b) the compressed encrypted image output from the first phase of the scheme; (c) the restored aerial image proceed from the second phase of the scheme with CR = 107 and PNSR = 21.55 dB

5 Conclusion

This paper presents a secured image compression system for image satellite communication based on the fractal theory of iterated contractive image transformation combined with the EHMC algorithm. The input aerial image passes through two main phases, encoding and decoding phases. Each phase has two stages. The first encoding stage compresses the input image based on the fractal algorithm giving high compression ratio. The second encoding stage encrypts the compressed input image by using the EHMC algorithm before transmitted it to the receiver at the decoding phase.

The first decoding stage is responsible for decrypting the compressed image and finally, the fractal decompression processes can decode at the last stage of the scheme. Although the compression ratio achieved is very high but the on the expense of some image quality performance. Improving the image quality while maintaining the high compression ratio of the restored image is under research. Also, it is shown that the cryptographic security of the reversed system is directly related to the strength of the key generator.

Acknowledgment

The authors wish to thank Prof. Dr. Shilong Ma in Beijing University of Aeronautics and Astronautics (BUAA), China. This work was supported in part by Dr. Ahmed Gohar, Egypt.

References

[1] J. Hutchinson, Fractals and self similarity, Indiana Unv. Math. J. , 2001.

[2] M. Barnsley, Alan D. Sloan, “A better way to compress images”; Byte, Jan 2008.

[3] A. E. Jacquin, “Image coding based on a fractal theory of iterated contractive image transformations”, IEEE Trans. On Image processing, Vol. 1, no. 1, Jan 2010.

[4] Y. Fisher, Fractal Image Compression – Theory and Application, Springer-Verlag, 2004.

[5] Sayood, Khalid, Introduction to Data Compression, Third Edition. Morgan Kaufmann. pp. 560–569. ISBN 0-12-620862-X, 2005.

[6] A. Aaron and B. Girod, Compression with Side Information Using Turbo Codes," in IEEE Data Compression Conference, April 2012.

[7] G. Cormode, M. Paterson, S. C. Sahinalp, and U. Vishkin, Communication complexity of document exchange," in Proceedings of the 11th annual ACM-SIAM symposium on Discrete algorithms, pp. 197-206, Society for Industrial and Applied Mathematics, 2011.

[8] M. Capalbo, O. Reingold, S. Vadhan, and A. Wigderson, Randomness conductors and constant-degree lossless expanders," in Proceedings of the 34th annual ACM symposium on Theory of computing, pp. 659-668, ACM Press, 2011.

[9] Mark Johnson et. al., On Compressing Encrypted Data, IEEE transactions on signal processing, vol. 52, no. 10, October 2004.

[10] A. D. Liveris, Z. Xiong, and C. N. Georghiades, Compression of Binary Sources with Side Information Using Low-Density Parity-Check Codes," in IEEE Communication Letters, 2002.

[11] Othman et al, “Enhanced Hill Multimedia Cryptosystem (EHMC)”, AEIC 2000, Proceedings of Al-Azhar engineering 6th International Conference, Vol. 9, pp. 135 - 140, Cairo, September 2009.

[12] J. Garcia-Frias and Y. Zhao, Compression of Correlated Binary Sources Using Turbo Codes," in IEEE Communication Letters, October 2010.