Non-linear Least Square method applied to Iterated Block Matching Fractals(IBMF) for image compression

Santos López
Advisor: Hamed Parsiani
Partially sponsored by NASA/PAIR-PaSCoR GRANT NCC5-340

Abstract

In this research, a non-linear least square (NLS) method is applied to an Iterated Block Matching Fractals(IBMF) software [1] which compresses an image. The present IBMF software uses affine transformations to find the parameters necessary to code an image block (Range block) in terms of a similar block from the same image (Domain block).

The NLS method was tested using C-language. The method consists of calculating non-linear least square parameters approximating a sample range block in terms of a similar domain block of the same image. The NLS software is embedded within the IBMF software and tested as compared with the IBMF's traditional use of affine transformations which calculates the needed transformations parameters for encoding of a Range block. The two methods are compared based on compression ratio and image quality, determined by Peak Signal to Noise Ratio (PSNR). The quality was improved, at the cost of lower compression ratio. This ratio will be determined as a future work.

1. Introduction

The non-linear least square (NLS) method [2] has been extensively used for different applications to approximate an unknown. Many images have self similarities within the same image which are exploited for compression of the same. In this paper, the NLS method is used to approximate a range block (BxB) of image in terms of another domain block of the same image. A second order non-linearity is applied which produces three variables representing a range block in terms of a domain block. The NLS method is being incorporated in the body of the IBMF image compression algorithm.

A. An example of NLS approximation

NLS algorithm is applied to a sequence of data and a non-linear curve is obtained which is the best curve fitting the data, see Fig.1.

Data to be fitted:
(X,Y)=[0,1;0.25,1.284;0.5,1.6487;0.75, 2.117;1,2.7183]

The non-linear best fit is:
$Y = 1.0052 + 0.8641X + 0.8437X^2$

Where coefficients are:
a = 1.0052, b = 0.8641, c = 0.8437

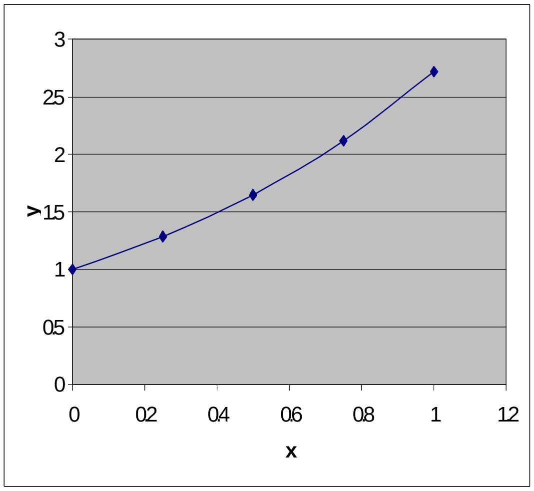

Fig.1 NLS curve fitting approximation

B. The concept of Iterated Block Matching Fractals

The concept of Fractals dates back to about 1982, when Mandelbrot observed that what we see around us is an intricate arrangement of features. If an image contains similar details, but at different scales, then it is a fractal image. Once these similarities are taken out then we have image compression.

The fractals as conceived by Barnsley [3][4], finds the basic features in an image, creates mathematical models of them, then defines the relations between pieces of the image and the basic features. One needs to know well the relationship between "formulas" for transformations and the resultant stretching, twisting, folding and skewing affecting the underlying image.

Arnaud Jacquin's work [5] on the principle of block matching does not rely on modeling of features by formulas. The method of Iterated Block Matching as introduced by Jacquin encodes the image using fractals similarities in the image. In this method, the image is broken down into equal size blocks (2Bx2B) called domains. Then an image block (BxB) to be encoded (called range block) is compared for the best match with each domain block (contracted to size BxB) from the domain pool. The matching parameters obtained are contrast scalings and luminance shifts. To increase the probability of finding a good match, different orientations (or isometries) of the domain blocks are created which produce a large pool of domain blocks with more varieties.

1. Contrast Scale Factor, Luminance Shift, and Approximate Range Equations

let Im = Image,

$D_j = j^{th}$ domain block,

$R_j = j^{th}$ range block

let $\mathrm{Im}|_{D_j} = j^{th}$ domain block

$\mathrm{Im}|_{SoD_j} = j^{th}$ contracted domain block

$\mathrm{Im}|_{\hat{R}_i} = j^{th}$ approximate range block

$\overline{\mathrm{Im}}|_{D_j}$ = Mean of D_j

$\mathrm{Im}|_{\hat{R}_i^{(n)}} = i^{th}$ estimated range block at n^{th} iteration

$\overline{\mathrm{Im}}|_{R_i}$ = Mean of R_i

$\mathbf{dr}|_{\mathbf{R_i}} \triangleq \mathrm{Im}|_{\hat{\mathbf{R}}_\mathbf{i}} - \overline{\mathrm{Im}}|_{\mathbf{R_i}}$ dynamic range of R_i

$\mathbf{dr}|_{\mathbf{D_j}} \triangleq \mathrm{Im}|_{\mathbf{D_j}} - \overline{\mathrm{Im}}|_{\mathbf{D_j}}$ dynamic range of D_j

Contrast Scale Factor $a_i \triangleq \dfrac{\mathbf{dr}|_{\mathbf{R_i}}}{\mathbf{dr}|_{\mathbf{D_j}}}$

Luminance Shift $\Delta g_i \triangleq \overline{\mathrm{Im}}|_{R_i} - a_i \overline{\mathrm{Im}}|_{D_j}$

<u>Approx. Range</u> is $\mathrm{Im}|_{\hat{R}_i} = a_i \mathrm{Im}|_{SoD_j} + \Delta g_i$

The coding of an image is done block by block in a raster scan fashion. Each block (a range) is coded by searching the pool of domains for the best match. The search time could be reduced by first classifying the image into blocks of the same type. Three block types are defined as: shade blocks, midrange blocks, and edge blocks. The domain pool is generated by sliding a window of size DxD (D=2B) across the original image in steps of B/2. Then the blocks are classified accordingly.

During the encoding procedure the image is scanned one block (range) at a time and the range block R_i to be encoded is compared against all domain blocks, and its isometries, of the same type and the best match D_j is obtained. From the best match the desired parameters a_i and Δg_i are calculated. The criteria for the best match is a distortion measure (RMS value) calculated for each $\mathrm{Im}|_{\hat{R}_i}$ and $\mathrm{Im}|_{R_i}$, where $\mathrm{Im}|_{\hat{R}_i} = a_i \mathrm{Im}|_{SoD_j} + \Delta g_i$.

2. Parameters Transmitted:

Shade Blocks:

Average gray level, $\overline{\mathrm{Im}}|_{R_i}$.

Midrange Blocks:

Average gray level, $\overline{\mathrm{Im}}|_{R_i}$, a_i, "type" of block, and the location of the matched domain (x,y).

Edge Blocks:

Same as the midrange blocks with an additional coding of the type of isometry. The code for isometry, I_i, is also transmitted.

3. Decoding Procedure

In the decoding process, at every iteration n, recursively a luminance shift Δg_i is generated using the received block means $\overline{\mathrm{Im}}|_{R_i}$ and the contrast scale factor a_i.

That is:

$\Delta g_i^{(n)} = \overline{\mathrm{Im}}|_{Ri} - a_i \mathrm{I_m}|_{\mathrm{Di}^{(n-1)}}$

Decoded Image Block,

$\mathrm{Im}|_{\hat{R}_i^{(n)}} = \mathrm{I_i}\{ a_i \mathrm{I_m}|_{\mathrm{SoDi}} + \Delta g_i^{(n)} \}$

The initial starting image at the decoder is a block average image which in the next 2 or 3 iterations will approximate the original image.

2. NLS implementation within the body of the IBMF method

A. NLS/IBMF encoder:

The NLS method calculates the mapping parameters, a, b, c, as in the previous example, where X is a 4x4 domain block and the Y is a 4x4 range block of an image to be compressed. The equations for these parameters are obtained by solving the simultaneous equations 1,2, and 3. These mapping parameters, their location and other overhead information must be coded and transmitted so that the decoder can rebuild the image. The block mean is transmitted as an overhead information. It turns out that we just need to transmit b, c and the a can be determined at the decoder.

$$a\sum_{i=1}^{16} X_i^0 + b\sum_{i=1}^{16} X_i^1 + c\sum_{i=1}^{16} X_i^2 = \sum_{i=1}^{16} Y_i X_i^0 \quad \text{Eq.1}$$

$$a\sum_{i=1}^{16} X_i^1 + b\sum_{i=1}^{16} X_i^2 + c\sum_{i=1}^{16} X_i^3 = \sum_{i=1}^{16} Y_i X_i^1 \quad \text{Eq.2}$$

$$a\sum_{i=1}^{16} X_i^2 + b\sum_{i=1}^{16} X_i^3 + c\sum_{i=1}^{16} X_i^4 = \sum_{i=1}^{16} Y_i X_i^2 \quad \text{Eq.3}$$

Approximate range is determined by:

$$\hat{Y}_i = a + b X_i^1 + c X_i^2 \quad \text{Eq.4}$$

A search of the best match as is done by IBMF produces the parameters a,b, and c.

B. NLS/IBMF Decoder:

At the decoder the, a, parameter is determined by Eq. 5.

$$\mathbf{a} = \mu_{\mathbf{y}} - \mathbf{b}\mu_{\mathbf{x}} - \mathbf{c}(\sigma_{\mathbf{x}}^{2} + \mu_{\mathbf{x}}^{2}) \qquad \text{Eq.5}$$

where $\mu_{\mathbf{y}}$ and $\mu_{\mathbf{x}}$ are the mean of the range and the domain block respectively and $\sigma_{\mathbf{x}}^{2}$ is the variance of the domain block.
A complete image is reconstructed using Eq.4.

3. Results

The results obtained from this method are in terms of the quality of the image measured by PSNR. From all the images tested higher PSNR were obtained than with the previous algorithm but at expenses of a lower compression ratio. For a satellite image from Añasco a PSNR of 37.4 was obtained that is better than the previous software which gives a PSNR of 37.1. The PSNR of Lena image was 34.4 and with the old method just a 33.7 was achieved. Results are shown in Fig. 2.

Fig.2(a) Original (512x512)

Fig.2(b) Reconstructed image PSNR = 37.4

References

[1] Hamed Parsiani, Andres Fuentes, "Fast Near lossless Iterated Block Matching Fractals Image Compression", Proceedings of the IASTED International Conference, Signal and Image Processing , SIP'98, Oct. 28-31, 98, Las Vegas, Nevada.

[2] Richard Burden, J. Douglas Faries, "Numerical Analysis," 1997, sixth edition, Brooks/Cole Publishing Co.

[3] M. F. Barnsley and S. Demko, "Iterated function systems, and global construction of fractals," Proc. Roy. Soc. London, vol. A399, pp. 243-275, 1985.

[4] M. F. Barnsley, "Fractals Everywhere". New York, Academic Press, 1988.

[5] A. E. Jacquin, "Fractal Image Coding: A Review," Proceedings of the IEEE, vol. 81, No. 10, Oct. 1993.