Relative fractal coding and its application in satellite image compression

S.K. Ghosh
Computer & Informatics Centre
IIT Kharagpur
skg@cc.iitkgp.ernet.in

J. Mukhopadhyay
Computer Science & Engineering
IIT Kharagpur
jay@cse.iitkgp.ernet.in

V.M. Chowdary
RRSSC / ISRO
IIT Kharagpur

A. Jeyaram
RRSSC / ISRO
IIT Kharagpur

Abstract

A new approach of relative fractal coding has been presented in this paper. In this technique given the fractal code of a reference image, one can generate a relative fractal code of any other image of same size. The convergence of the relative code is also guaranteed. This method found to be useful for satellite remote sensing image compression as the spectral bands are correlated. It produces pure fractal code where the inter-band spectral changes are coded and the convergence is guaranteed for individual bands. Several experimental results are also presented.

1. Introduction

Fractal image compression has been drawing considerable attentions of the researchers since Barnsley [1] [3] has outlined a schema based on Iterated Function System (IFS) (and its other variations) for a potentially high compression scheme. Later, Jacquin [5] [2] came up with an algorithm based on Partitioned Iterated Function System (PIFS) for encoding an image.

In recent time the satellite remote sensing data [1] is extensively used for natural resource mapping/monitoring, disaster management etc. There is a growing need for satellite image compression as these data occupy considerable disk space. Conventional fractal compression schemes can easily be extended to satellite image compression as a satellite image is usually represented in multi-band. Thus each band in satellite image can be compressed as a grey-level image.

In this paper a new approach for fractal image coding based on an innovative concept of *relative fractal coding* has been proposed which found to be suitable for coding multi-band satellite image. This scheme produces a pure fractal code where the local inter-component spectral changes are coded and the convergence is guaranteed for individual band.

[1]In this paper optical remote sensing data is considered. The satellite remote sensing image is also referred as *satellite image* in the rest of the text.

2. Overview of Fractal Image Encoding and Decoding

In fractal compression an image is encoded as the attractor of an iterated function system. It is based on the observation that natural images are partially self-transformable [5]. They contain 'affine redundancy' in the sense that a block in the image (called *range*) can be derived from another block of the same image(called *domain*) by some affine transformation. In the fractal encoding method, following Jacquin's approach, the encoding process starts with partitioning of the image into a set of non-overlapping segments(*range blocks*) and then for each range block an image block (*domain block*) with different resolution is searched that gives the best affine mapping to the range segment. Compression is achieved by encoding the location of the domain blocks and the affine transformation for each range segment.

2.1 The Conventional Algorithm

The conventional algorithm as proposed by Jacquin [5] can be summarized as follows. An image I is defined as the mapping of points in discrete 2D space $Z \times Z$ to grey level values belonging to real set R such that $I : Z \times Z \rightarrow R$. The input image I is partitioned into non-overlapping *range blocks*, each of size $R\_Size \times R\_Size$ and overlapping *domain blocks* of size $D\_Size \times D\_Size$. This has been illustrated in figure 1. For every range block R_i in I, a suitable *domain* block D_j in I is located by exhaustively searching the image and an associated affine transformation W such that R_i can be reconstructed (at least approximately) as $W(sh(iso(D_j)))$, where $iso(.)$ denotes isometric transformation on D (say rotation) and $sh(.)$ is a shrinking operation (say averaging or decimation operations). The criteria for selecting the domain D_j for range R_i is that the error between R_i and $W(sh(iso(D_j)))$ (i.e. the *Root_Mean_Square(RMS) error* is less than a threshold value. It may be noted that this threshold is termed in this work as the *qualifying block threshold* (QBT). The lattice separation between two successive domain blocks are represented by l. The experimental results presented in this paper are with QBT and l values as 10 and 1 respectively (if not otherwise mentioned).

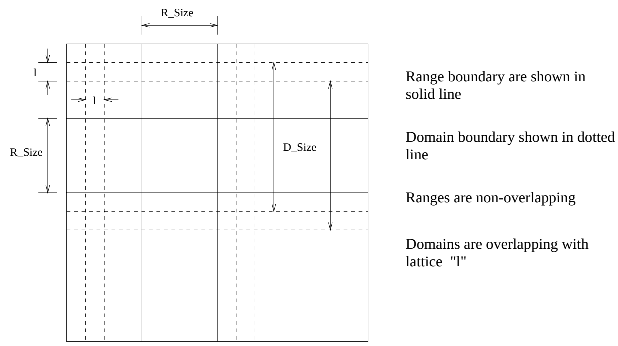

Figure 1: Encoding Parameters

In the compressed image, the $< D, sh, iso, W >$ information is stored for every range block. As the shrinking function sh will be uniformly applied to each domain, it is sufficient to store only $< D, iso, W >$. As W is affine this can be expressed by $< a, b >$ such that: $W : y = ax + b$, where $x, y \in R$ and two constants a and b are also in R. In this case a is known as *scale factor* and b is known as the *translation factor* of the affine transform. The *conventional algorithm* is also referred as *exhaustive approach* as for each range block exhaustive search of domain blocks is performed.

2.2 Fractal decoding

The decoding process is considerably simple. Given the codes for a range R_i in the form of $< D_j, iso_i, W_i >$ one has to iteratively apply $W_i((sh(iso_i(D_j))))$ for every i^{th} range block. The convergence of the decoder is guaranteed when the magnitudes of the scale factors (a's) are kept less than *one*. In [6] it has been shown that if the decimation is used as shrinking operation in the above algorithm, the convergence may be obtained even if the magnitudes of a few of the scale factors are kept below one. The rest of the scale factors may not follow the above restriction.

2.3 Convergence of the decoder and the concept of *limit cycle*

A graph-theoretic interpretation of convergence of fractal encoding based on Partial Iterated Function System (PIFS) is presented in our previous work [6]. The concept leads to the development of a linear time fast decoding algorithm from the compressed image.

It has been shown in [6] that the encoded image can be modeled as a *flow graph*, where the directed edges from a pixel q to p exists if the brightness value of the pixel p is determined from q. It is proved that *if for every pixel there exists one and only one pixel from which the brightness values are computed, the image space is partitioned into a set of* circular plants. The structure of a circular plant consists of a circular chain, called as its *limit cycle* and several chains of pixels coming out of the *limit cycle*. Each circular plant converge independently. The convergence of the encoder under this circumstance depends upon the convergence of these *limit cycles*. A chain is called a σ*-chain* when it has a *limit cycle* or it is *circular*. The example of a σ*-chain* and a *circular plant* are shown in figure 2. The detailed discussion and analysis of this partitioning could be found in [6]. The concept also leads to the development of a linear time fast decoding algorithm.

There are a few other important observations drawn from this study [6], namely,

- The encoded image is partitioned into a set of circular plants, each of which is *convergent if its limit cycle converges*

- The scale factors of the affine transforms corresponding to the points lying on *the limit cycles* should be between -1 and 1. For others, they may assume arbitrary values. As a result, only a very small fraction of the affine transforms require to satisfy the constraint of keeping the magnitude of the scale factor below 1.

- The fraction of points lying in the limit cycles is extremely small.

3. Relative fractal coding

In this section the concept of *relative fractal coding* has been introduced. Using *relative fractal coding* scheme fractal codes for the individual bands of a satellite image are generated and at the same time the convergence of the decoder is also guaranteed. In this encoding technique, given the fractal code of a reference image, one can generate a relative fractal code of any other image of the same size. This relative fractal code combined with the code of the reference image, produces the complete fractal code of the target image. More the similarity in the images, less is the size of the relative code. In the relative code, the same range-domain mappings of the reference image are used. Only the transformation of brightness values are changed, if required.

The overview of the conventional fractal encoding and decoding methods is given in the section 2.1. With the help of the notations used earlier for describing the compression algorithm, the principle behind the relative fractal coding is explained here. Let the ith range R_i in the reference image I_r be encoded as $< D_j, iso_i, W_i >$. For another image I, one may use the same range partitioning and the same *range-domain mapping* as well as the *isometry*, for

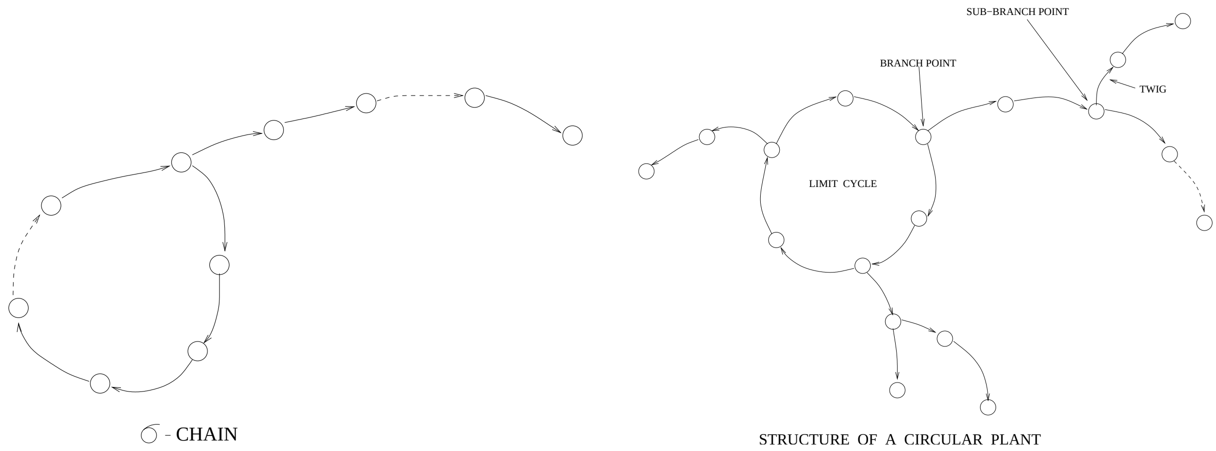

Figure 2: σ-chain and *circular* plant

encoding it. In this case, the affine transform W_i has to be newly computed for minimizing the error between $I(R_i)$ and $I(sh(iso_i(D_j)))$. Let the new transform be found as W_i' for encoding the ith range of I. Hence given the reference fractal code of I_r, one requires to compute and store the affine transforms for every range of an arbitrary image. This is known as the *relative fractal code* of I with respect to I_r. It may be noted that as the mapping between the range and domain blocks are already defined for the relative code, the encoding speed is much faster than the usual fractal encoding algorithm.

3.1 Convergence of *relative fractal code*

One of the problems of designing a fractal code is to ensure the convergence of the decoder. Hence, given a fractal code of a reference scene, it is difficult to bring any modification in the code for adapting the local changes of the reference scene. This is because any arbitrary changes in the transformations may not guarantee the convergence. But in this work by computing *limit cycles* (refer section 2.3) during encoding [6], the convergence problem has been resolved. *As the convergence of the fractal decoder solely depends on the convergence of the limit cycles, given the brightness values of the limit cycle points, decoding process of the fractal codes will always converge. It may be noted that as the number of limit cycle points is very small, this will add an insignificant amount of overhead on the fractal code size.* Hence the respective brightness values of the points of the limit cycles are appended with the relative code (refer step 2 of the algorithm *Relative_Fractal_Coding* in the next section). A detailed study of the convergence through computation of *limit cycles* is presented in [6].

3.2 Algorithms

The algorithms for the relative fractal coding is described below.

Algorithm : **Relative_Fractal_Coding**
Input : An image I, The reference fractal code.
Output : The relative code of the image I.

Begin

Step 1. For every range block R do $\{$

Get the reference code for the range as $< D_R, iso_R, W_R >$.

Find the affine transform W_R' which minimizes $E(I(R), W_R'(I(sh(iso_R(D_R)))))$.

Output $< W_R' >$.

$\}$

Step 2. Compute the sequence of limit cycle points and output their brightness values in I.

End **Relative_Fractal_Coding**

While decompressing the image the brightness values at the limit cycle points of the encoded stream are directly used for ensuring the convergence of the decoder. The decoding algorithm is described below.

Algorithm : **Relative_Fractal_Decoding**
Input : Fractal code of the reference image I_R, Relative fractal code of I.
Output : Decoded image I_d.

Begin

Step 1. Form the full fractal code of I as follows : For each range block R_i get the domain address (D_j) and isometry information (iso_j) from the fractal code of the reference image I_R and get the affine transformation (W_i') from the relative fractal code of I. This will produce the triplet $< D_j, I_R, W_i' >$ for every ith range block in I.

Step 2. Compute the limit cycle points and assign the brightness values to those pixels (in the reconstructed image I_d) obtained in the same order from the encoded stream of the relative fractal code of I (refer step 2 of the algorithm Relative_Fractal_Coding).

Step 3. Apply non-iterative decoding process for the rest of the pixels while reconstructing the image I_d. It may be noted that the values at the limit cycle points are already obtained from the encoded stream in the previous step.

End **Relative_Fractal_Decoding**

The relative fractal coding is basically meant for images having considerable similarity with the reference image. As it is expected that many of the range blocks in an image remain (almost) unchanged with respect to the reference image, the associated affine transforms of the reference image also could be used in the relative codes. Hence in this case only a few number of affine transforms are required to be stored with an additional overhead of storing the associated range information. The algorithm for the modified relative fractal coding is presented below. Here the affine transforms of the reference image are used in the relative codes if root mean square (RMS) error is within a particular threshold. This threshold is referred here as *Relative_error_threshold*.

Algorithm **Modified_Relative_Fractal_Coding**
Input : An image I, Reference fractal code, Relative_error_threshold.
Output : The relative code of the image I.
Begin

For every range block R do $\{$

Get the reference code for the range as $< D_R, iso_R, W_R >$.

Find the affine transform W_R' which minimizes $E(I(R), W_R'(I(sh(iso_R(D_R)))))$.

Calculate error difference,
$Err\_dif = E(I(R), W_R'(I(sh(iso_R(D_R))))) - E(I(R), W_R(I(sh(iso_R(D_R)))))$.

If $(Err\_dif < Relative\_error\_threshold)$ $\{$

Output the range identity and $< W_R' >$.

$\}$

$\}$

Compute the sequence of limit cycle points and output their brightness values in I.

End *Modified_Relative_Fractal_Coding*

4. Satellite image compression using *relative fractal coding*

The concept of relative fractal coding is found to be useful in compressing satellite images as the spectral bands are correlated. In a satellite image as its spectral components are strongly correlated, one of them could play the role of the reference image. The other two components are relatively coded with respect to the code of the reference one. The selection of the component as the reference image is important. Out of the three bands (Bands 1, 2, 3) of the optical remote sensing data, the *Band 2* is selected as reference image as this shows maximum co-relation with the other two.

4.1 Algorithms for fractal compression of satellite image

The algorithms for fractal coding of satellite images are presented in here.

Algorithm : **Sat_Image_Compression**
Input : Satellite Image with Band 1 (B1), Band 2 (B2), Band 3 (B3).
Output : Fractal codes.
Begin

Step 1. Encode the B2 with the conventional fractal coding technique.

Output the code of B2.

Step 2. Perform relative coding of the B1 and the B3 bands with respect to the B2 band.

Output the relative codes of B1 and B3 bands.

End *Sat_Image_Compression*

During decompression first the full fractal codes of the *Band 1* and the *Band 3* components are recovered (from the fractal codes of the *Band 2* component) and then each component is reconstructed independently from its code.

4.2 Experimental results

The results obtained for satellite image compression by these methods are encouraging. The experiments are carried out in the Pentium III (866 MHz, RAM 256 MB) system under LINUX 7.0 operating system. The *range* size are kept as $8x8$ and the values of *qualifying block threshold* (QBT) and *relative_error_threshold* are kept as 10 and 2 respectively. The verification of the data is carried out using ERDAS/IMAGINE (ver 8.3) image processing software.

Two satellite images (size $512x512$), namely *SatA* and *SatB* are taken for experimentations. These are optical remote sensing data (Band 1, 2, 3) of IRS-1D (LISS III) satellite. The images are both individually band-wise fractal-coded and relatively coded with *Band 2* as reference. The results are presented in tables 1 and 2 respectively.

Image	Band 1		Band 2		Band 3	
	BPP	PSNR (in dB)	BPP	PSNR (in dB)	BPP	PSNR (in dB)
SatA	0.57	35.46	0.57	34.15	0.57	33.24
SatB	0.58	35.52	0.58	33.93	0.58	33.68

Table 1: Satellite image compression with individual bands fractal-coded

Image	BPP	PSNR (in dB)
SatA	1.20	33.15
SatB	1.20	32.71

Table 2: Satellite image compression using *relative fractal coding* with *Band 2* as reference

In order to measure the effect of degradation in the reconstructed image quality following method has been adopted.

- Supervised classification (using *maximum likelihood classifier*) [4] of the original satellite data is carried out with a given *signature set*.
- The same signature set is used to classify the reconstructed images.

The classification results of images *SatA* and *SatB* are shown in tables 3 and 4 respectively. The images are classified into several classes, namely, Sand, Water, vegetation, Fallow, Waste land, and the area covered (in percentage) are shown in the tables.

It can be observed from the tables 3 and 4 the results are closely matching. It may be noted here that the classification is carried out only for broad classes, not much stress has been given to the classification accuracy.

Class	Original data	Individual band fractal-coded	Relative fractal coding
Sand	3.38%	3.25%	3.30%
Water	14.96%	12.84%	11.71%
Vegetation	50.34%	58.05%	57.20%
Fallow	31.31%	25.86%	27.80%

Table 3: Classification results of image *SatA*

Class	Original data	Individual band fractal-coded	Relative fractal coding
Water	42.07%	42.15%	40.71%
Vegetation	41.96%	44.48%	44.88%
Fallow	15.43%	13.04%	14.09%
Waste land	0.55%	0.32%	0.32%

Table 4: Classification results of image *SatB*

The classification results along with the standard FCC (False color composite) are shown in figures 3 and 4.

5. Conclusion

A new method for fractal coding for satellite images has been proposed is this paper. It is based on an innovative concept of *relative fractal coding*. The major problem in fractal coding using the inter-band spectral changes, is to ensure the decoder convergence. In this technique the convergence of the decoder has been ensured by computing the *limit cycle points* and transferring those points while coding the spectral bands with reference to other.

References

[1] M. Barnsley. *Fractals Everywhere*. Academic Press, 1988.

[2] M. Barnsley and A. Jacquin. Application of recurrent iterated function system to images. *SPIE Visual Communications and Image Processing*, 1001:122–131, 1988.

[3] M. Barnsley and A. Slaon. A better way to compress image. *BYTE Magazine*, January 1988.

[4] R. Gongalez and R. Woods. *Digital Image Processing*. Addison Wesley, 1993.

[5] A. E. Jacquin. Image coding based on a fractal theory of iterated contractive image transformations. *IEEE Transaction on Image Processing*, 1, January 1992.

[6] J. Mukherjee, P. Kumar, and S. Ghosh. A graph-theoretic approach for studying the convergence of fractal encoding algorithm. *IEEE Transaction on Image Processing*, 9, March 2000.

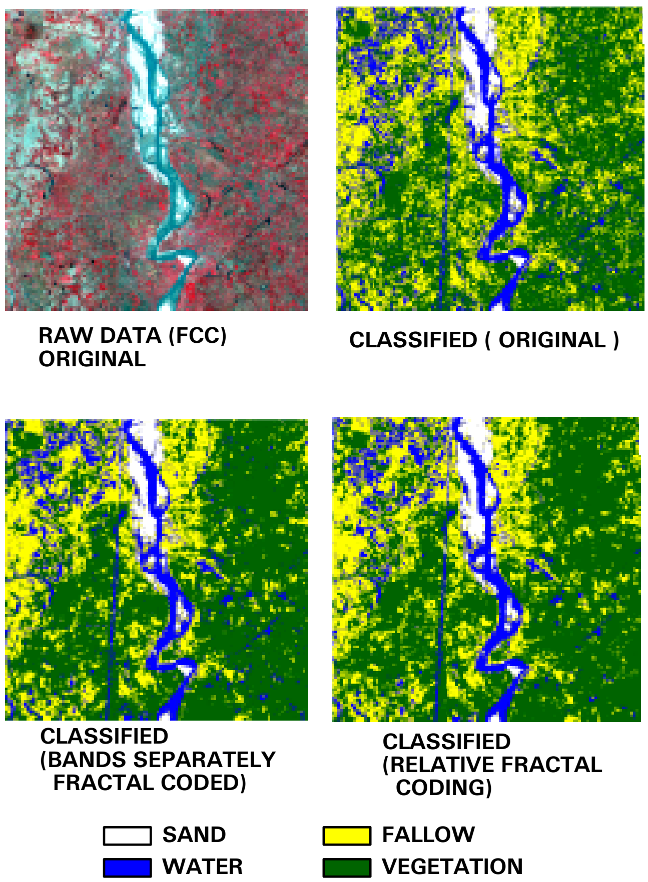

Figure 3: Classification results of *SatA* image

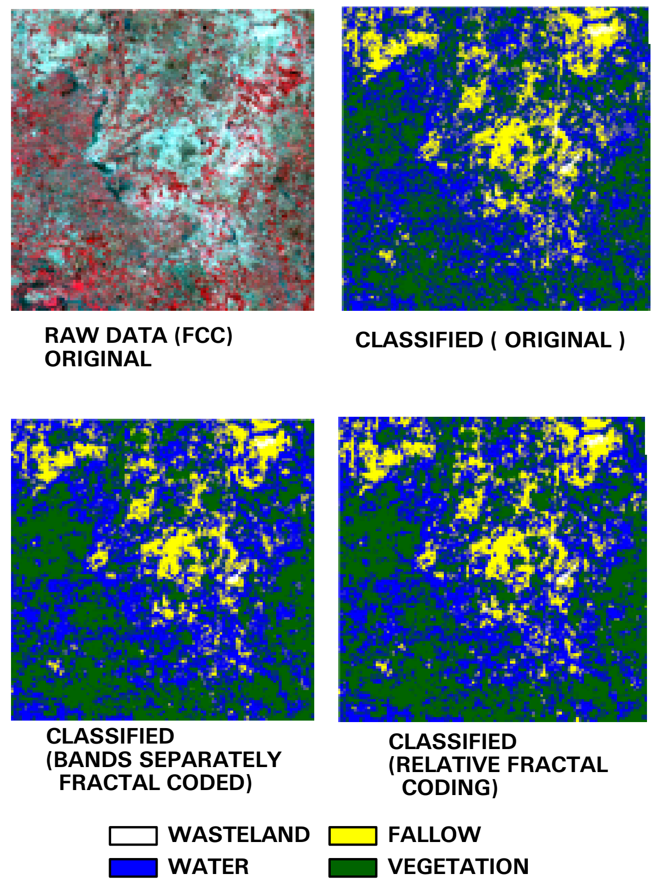

Figure 4: Classification results of *SatB* image