A STUDY ON IMAGE COMPRESSION WITH PARTICULAR REF‸ERENCE TO MEDICAL IMAGE

By

Saumitra Kayal

Under the supervision of

Prof. Dwijesh Dutta Majumder

Indian Statistical Institute
203 Barrackpore Trunk Road
Calcutta 700 - 035

July 1998

Dedicated

to

all of them

who stand as

my constant source of inspiration.

ACKNOWLEDGEMENT

I would like to express my sincere most gratitude towards my guide Prof. Dwijesh Datta Majumder for his invaluable guidance inspiration , encouragement and guidance in every step of carrying out this project .

I am also extremely grateful to Mrs. Mahua Bhattacharjee for technical help me in every step to developed this project in lab.

I would like to thank Mr. Subhabrata Mukherjee , for providing help in typing to this report.

Saumitra Kayal 29/7/98

Saumitra Kayal.

ABSTRACT

Different conventional and Fractal base image compression techniques will be applied on biomedical images and comparative study of results will be attempted.

Here we developed Quadtree based fractal compression method and also discuss the merging procedure of wavelet and fractal compression method .

CERTIFICATE

The forgoing thesis is hereby approved as a creditable study of an engineering subject (M. Tech Computer science) carried out and presented in a manner satisfactory to warrant its acceptance as pre-requisite to the degree for which it is submitted . It is understood that by this approval the undersigned do not endorse or approve any statement made , opinion expressed or conclusion drawn therein but approve the thesis only for which it is submitted .

Signature of examiners

CONTENTS PAGE NO.

CHAPTER 1.

1. Introduction :

Data compression has become an important issue for information storage and transmission. This is especially true for databases consisting of a large number of detailed computer images .Recently, many methods have been proposed and developed for achieving high compression ratios for compressed image .A promising compression technique, in terms of compression ratios, is fractal image compression . Here we use fractal compression on wavelet transformation of a grayscale image. The advantage of hybridising the wavelet and fractal is not just for pure compression performance , in fact, there are two important features that are strongly demanded in today's digital image market : ***progressiveness*** and ***resolution independence*** . As a matter of fact there are technologies that can achieve either feature well but not both, until now . The wavelet sub-band coding technology has a natural progressive structure that has been proven by Said and Pearlman [2]. And the fractal image representation gives a powerful resolution-independent image representation that serves for multi-uses.

We apply this method on medical images, since medical images typically have a degree of randomness associated with both the natural random nature of the underlying structure . If we regard the pixel intensity as high above a plane , then the intensity surface of a medical image can be viewed as a rugged surface . The fractal concept provides an excellent explanation of the ruggedness of natural surfaces and many other natural phenomena.[3] .

Now the question is what advantage does a wavelet compression system have that other compression techniques do not ? The essence of wavelets is variable manipulation . As a result , the gains are reflected in two geometric aspects. The first gain is in decorrelating the ***vertical*** and ***Horizontal***, and ***diagonal*** frequencies of an image locally . Indeed , most image pieces have definite pictures in one of the vertical, horizontal, or diagonal direction .Actually ,when an image piece significantly involves all directions, it would considered a noisy pieces, where higher distortion would be acceptable . The second geometric gain is derived from the delicate choice of wavelet [1],[2].

Here we Choose Haar-Walsh-Hadamard decomposision which is equivalent to the square image partition ; Its lossy image representations always give a blocky images , Which are visually intolerable.

1.1 What is Fractal Image Compression ?

Imagine a special type of photocopying machine that reduces the image to be copied by a half and reproduce it three times on the copy, as in figure . What happens when we feed the output of this machine back as input ? which we show here by applying several process of this method . Ultimately we will get a fixed image , we call this image as ***attractor*** [1].

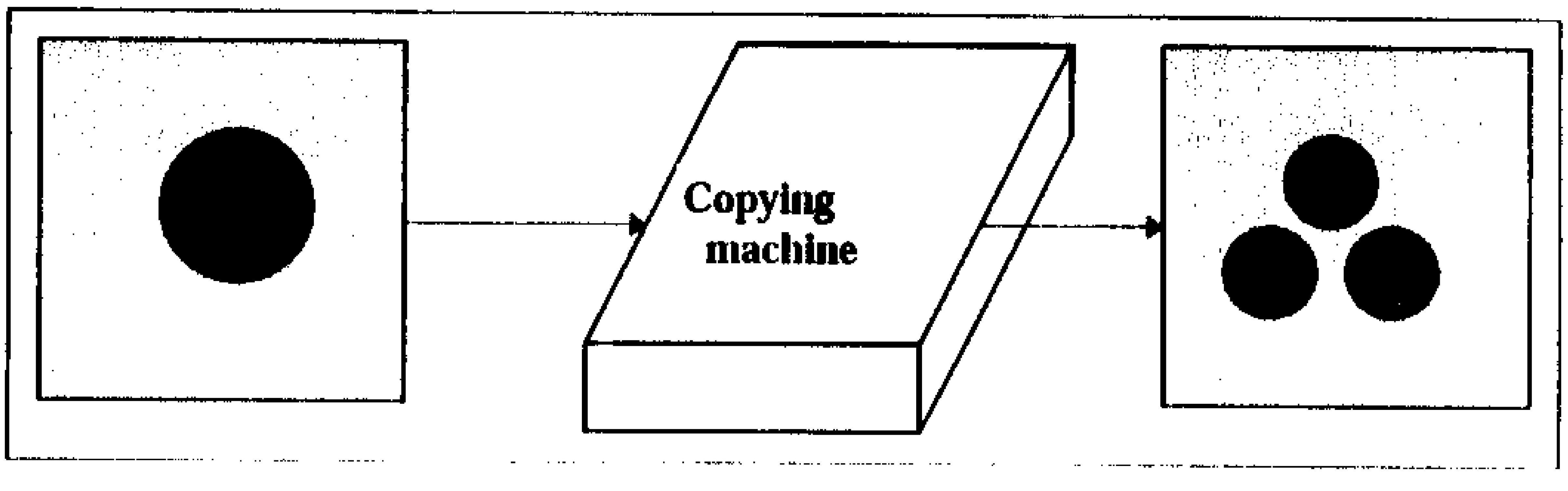


Fig :1 ***Copying machine generates three copies of the circle image .***

Since the final result of running the copy machine in a feedback loop is determined by the way the input image is transformed ,we only describe these transformations. Different trans- formations lead to different ***attractors,*** with the technical limitation that the transformations must be ***contractive***—that is ,a given transformation applied to any two points in the input image must bring them closer together in the copy. This technical condition is very natural since if points in the copy were spread out ,the attractor might have to be of infinite size.Except for this condition ,the transformations can have any form. In practice, choosing transformation of the form

$$w_i\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} a_i & b_i \\ c_i & d_i \end{bmatrix}\begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} e_i \\ f_i \end{bmatrix}$$

is sufficient to yield a rich and interesting set of attractors . Such a transformation are called Affine Transformations of the plane and each can skew ,stretch, rotate, scale and translate an input image.

1.2 Previous works :

The term fractal was first used by Benoit Mandelbort to designate objects that are self similar at different scales . Mandelbort did not actually consider fractals for compression , but he showed that they could be used for modeling real objects such as clouds , trees , mountains ,etc. Micle Barnsley and his co-workers at the Georgia Institute of Technology were first to recognize the potential interest of fractal method for image compression . For this, Barnsley developed the theory of *Iterated Function System* (IFS) in 1981 . But this method comes to impractical for lots of calculation and also it is not an automated method . In 1988 Jacquine gives an idea about partition the image in non overlapping region and Y .Fisher , Roger Boss and Bill Jacobs developed a theory which is known as Partition IFS . In 1995 Y Fisher developed the algorithm Quadtree based fractal compression also lots of research also going on about this method .

1.3 Our Work :

We have developed the Quadtree based fractal compressin method using Y Fisher 's algorithm . We have tested our algorithm on medical image and also try to minimize the loss of the image for a given small threshold value . Since the image is medical we always try to keep the loss minimum . After completion of this method we try to implement a merging method of fractal and wavelet . Our objective is to compress an image data as much as possible and simultaneously check for minimum loss of the image .

CHAPTER 2 :

2. Mathematical Background :

Hutchison introduced the theory of iterated function system (a term coined by M. Barnsley) to model collection of contractive transformations in a metric space as dynamical systems . His idea was to use the ***Contractive Mapping Fixed-Point Theorem*** to show the existence and uniqueness of fractal sets that arise as fixed points of such systems . It was barnsley s observation , however , that led to the idea of using iterated function systems (IFS s) to encode the images. He noted that many fractals that can be very compactly specified by *Iterated Function Systems* have a natural appearance. [1]

2.1 Fractals

There is no hard and fast definition of fractals , but has a list of properties . If we consider a set F to be fractal , we think of it as having some of the following properties:

1. ***F*** has detail at every scale.
2. **F** has (exactly, approximately, or statistically) self-similarity.
3. The *fractal dimension* of ***F*** is greater than its *topological dimension* .
4. There is a simple algorithmic description of ***F.***

Definition 2.1: The topological dimension of a totally disconnected set is always ***zero*** . The topological dimension of a set *F* is **n** if arbitrary small neighbourhoods of every point of ***F*** have boundary with topological dimension **n** - **1**.

2.2 Complete Metric Space:

Let $I(w)$ be an interval of real line $(0,w)$ for some $w > 0$. An image f is a function $f{:}I(w)xI(h) \rightarrow I(i)$, where w,h,i are width, height, and maximum intensity value of the image , respectively. For simplicity ,we assume $w = h$, and let $I(w) = I(h)$. Thus, the image can be viewed as a set in $I\ xI(i)$. By imposing some distance metric d on $I\ xI(i)$ we create a metric space X.

Definition 2.2: A metric space is a set X on which a real- valued distance function d: $XxX \rightarrow R$ is defined , with the following properties :

1. $d(a,b) \geq 0$ for all a , b $\in X$.
2. $d(a,b) = 0$ if and only if a = b, for all a, b $\in X$.
3. $d(a,b) = d(b,a)$ for all a,b $\in X$
4. $d(a,c) \leq d(a,b) + d(b,c)$ for all a,b,c $\in X$

such a function d is called metric .

Definition 2.3: A sequence of points $\{x_n\}$in a metric space is called a Cauchy sequence if for any $\in > 0$, there exists an integer N such that

$$d(x_m,x_n) < \in \text{ for all } n, m > N$$

Definition 2.4: A metric space X is complete *if* every Cauchy sequence in X converges to a limit point in X.

Definition 2.5 : A metric space X is compact if it is closed and bounded.

Definition 2.6: A contractive mapping W on a metric space X with metric d, is a mapping $W: X \rightarrow X$ such that for any two points x,y $\in$ X we have

$$d(W(x),W(y)) \leq s.\ d(x,y) \quad \text{where } s < 1.$$

2.2.1 The Contractive Mapping Fixed Point Theorem:

Let X be a complete metric space and f : X →X be a contractive mapping . Then there exists a unique point $x_n \in X$ *such that for any point* $x \in X$

$$x_n = f(x_n) = \lim_{n \to \infty} f^{on}(x)$$

such a point is called a fixed point or attractor of the mapping f.

proof: proof given in reference [1].

2.2.1 Corollary : (Collage Theorem) : With the hypothesis of the contractive Mapping Fixed Point Theorem.

$$d(x,x_n) \leq 1/(s-1) * d(x, f(x))$$

2.2.2 Generalised Collage Theorem :

Let f be eventually contractive with exponent n; then there exists a unique fixed point $x_n \in X$ *such that for any* $x \in X$

$$x_n = f(x_n) = \lim_{k \to \infty} f^{oK}(x)$$

in this case

$$d(x, x_n) \leq 1/(1 - s) * (1 - \sigma^n / 1 - \sigma) * d(x, f(x)),$$

where s is the contractivity of f and σ is the Lipschitz factor of f.

Proof: given in reference [1].

2.3 Iterated Function System :

Before discussing about iterated function system we first try to understand about the generation of synthetic image using fractals. Consider three transformations as follows;

$$W_1\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 1/2 & 0 \\ 0 & 1/2 \end{bmatrix}\begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

$$W_1\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 1/2 & 0 \\ 0 & 1/2 \end{bmatrix}\begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

$$W_1\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 1/2 & 0 \\ 0 & 1/2 \end{bmatrix}\begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

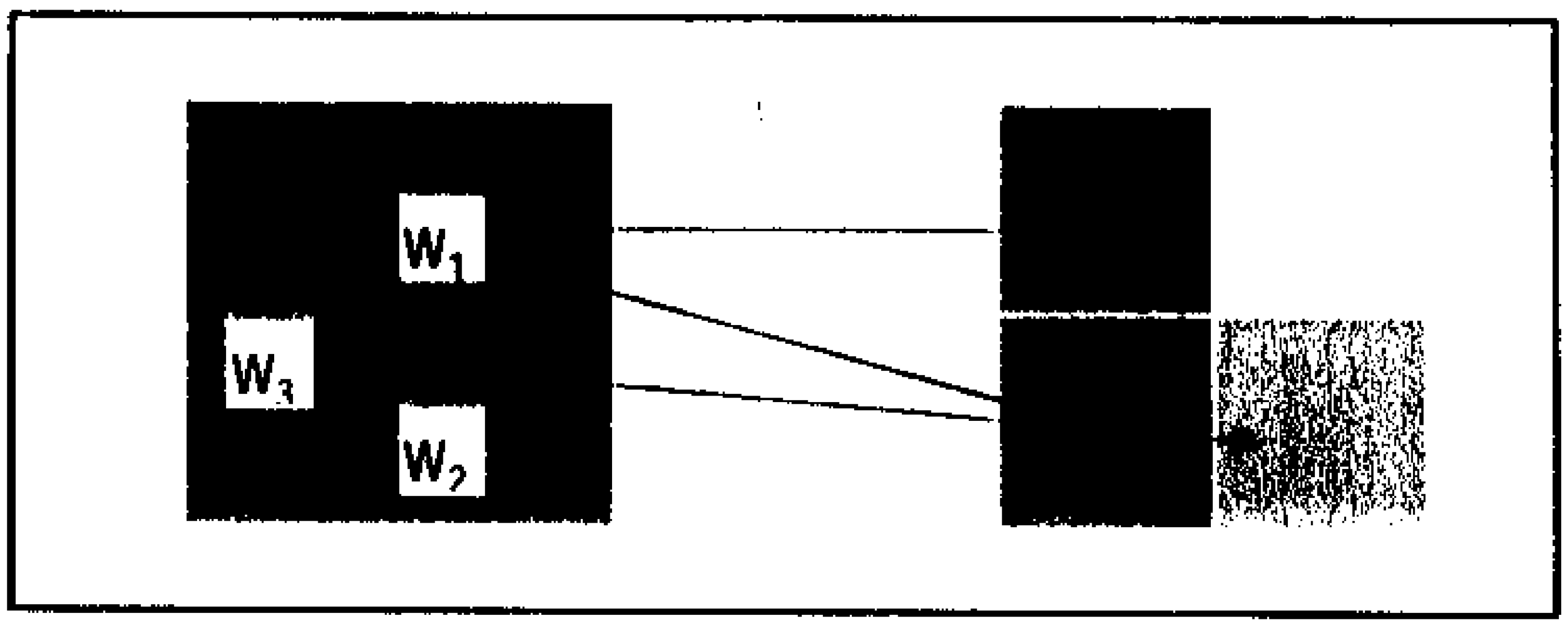


fig: 2 ***for three maps w_1, w_2, w_3***

For any set S, let

$$W = \bigcup_{i=1}^{3} w_i$$

Denote the n-fold composition of W with itself by W i.e W = W(X) W = W(W(X)) and so on

$$W = \underbrace{W(W(W(........W(X))))}_{k \text{ number}}$$

and let $I = [0,1]$. Now define $X = \{(x,y):0 < x < 1, 0 < y < 1\} = I^2$ and $A^k = W(A^{k-1})$. Then as $k \rightarrow \infty$ the set A^k converges to a limit set A^∞ .In fact, for any compact set $S \subset R^2$, $W^{On}(S) \rightarrow A^\infty$ .as $n \rightarrow \infty$. This A is called Attractor (or fixed point) in the image space. I.e. W(A) = A .

Fig : 3.1

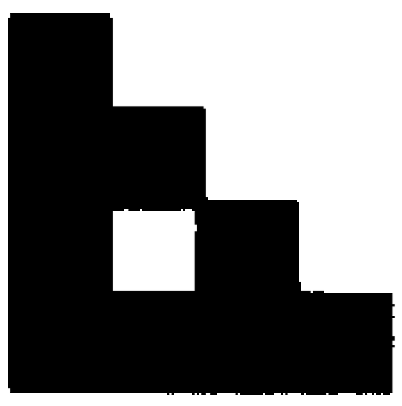


Fig 3.2

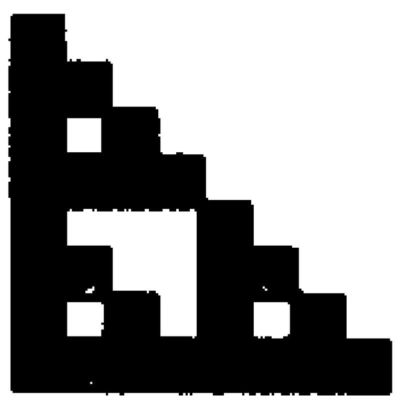


Fig 3.3

Figure 2 corresponds to map W applying on figure 1 and fig : 3 corresponds to map W applying on fig 2 .

So an iterated function system (IFS) on X is the union of a collection of contractive mappings {Wi} on metric space X . According to the contractive mapping fixed point theorem, stated as follows:

If F is a complete metric space, and W:X→X is a contractive transformation, then there exits a unique fixed point

$$|W| = \lim_{n \to \infty} W(A) \text{ , for any } A \in X.$$

An IFS has a unique attractor in the space of possible images on I .

This formula suggest that we will seek the transformations w_1, w_2, w_3 , ... w_n which encode a given fractal .The transformations should satisfy the relation

$$|W| = W(|W|) = w_1(|W|)U....U\ w_n(|W|).$$

which means that the fractal |W| is built up of parts that are the result of compressing ,rotating, translating, and skewing the fractal itself. The parts $w_1(|W|)$,.... $w_1(|W|)$are said to cover the fractal |W|.

But this process can t be automated, the machine can find the good IFS only with considerable help from a person guiding the compression process. Thus IFS based compression turned out impractical.

2.4 Partitioned iteratted function system (PIFS):

A partitioned IFS on a metric space is a collection of of contractive mappings with restricted domains. Let $D_1, D_2, \ldots, D_n$ be subsets of X and $R_1, R_2, \ldots, R_n$ be a disjoint cover of I. We refer to the collection {Di} as the set of domains and {Ri} as the set of ranges. A PIFS W on X is the union of a set of contractive mappings {Wj} such that for each k {1,2, m} there exists a j such that

$$Wj : Dj \times I(I) \rightarrow Rk \times I(I).$$

Such a PIFS satisfies the contractive mapping fixed-point theorem, and thus has a unique attractor f in the space of possible images on I.

Note : The objective of fractal image compression is to find a PIFS whose attractor is very close to the original image. Image compression realised when the storage requirements of the PIFS are lower than the original image.

2.5 Encoding Images With PIFS.

Definition 2.: *If $w : R^3 \quad R^3$ is a map with $(x',y',z1') = w(x,y,z1)$ and $(x',y',z2') = w(x, y, z2)$, then w is called z contractive if there exists a positive real number $s < 1$ such that*

$$| z1' - z2'| \leq s | z1 - z2 |$$

and if x' and y' are independent of z1, or z2 for all x, y, z1, z2.

Lemma : If w_1, w_2 , .., w_n are z-contractive , then

$$W = \bigcup_{I=1}^{n} w_i$$

is contractive in X with the supremum metric .

Proof : given in reference [1].

2.6 Quadtree Partitioing Scheme:

A quadtree partition is a representation of an image as a tree in which each node , corresponding to a square portion of the image , contains four subnodes , corresponding to the four quadrants of the square . The root of the tree is the initial image .

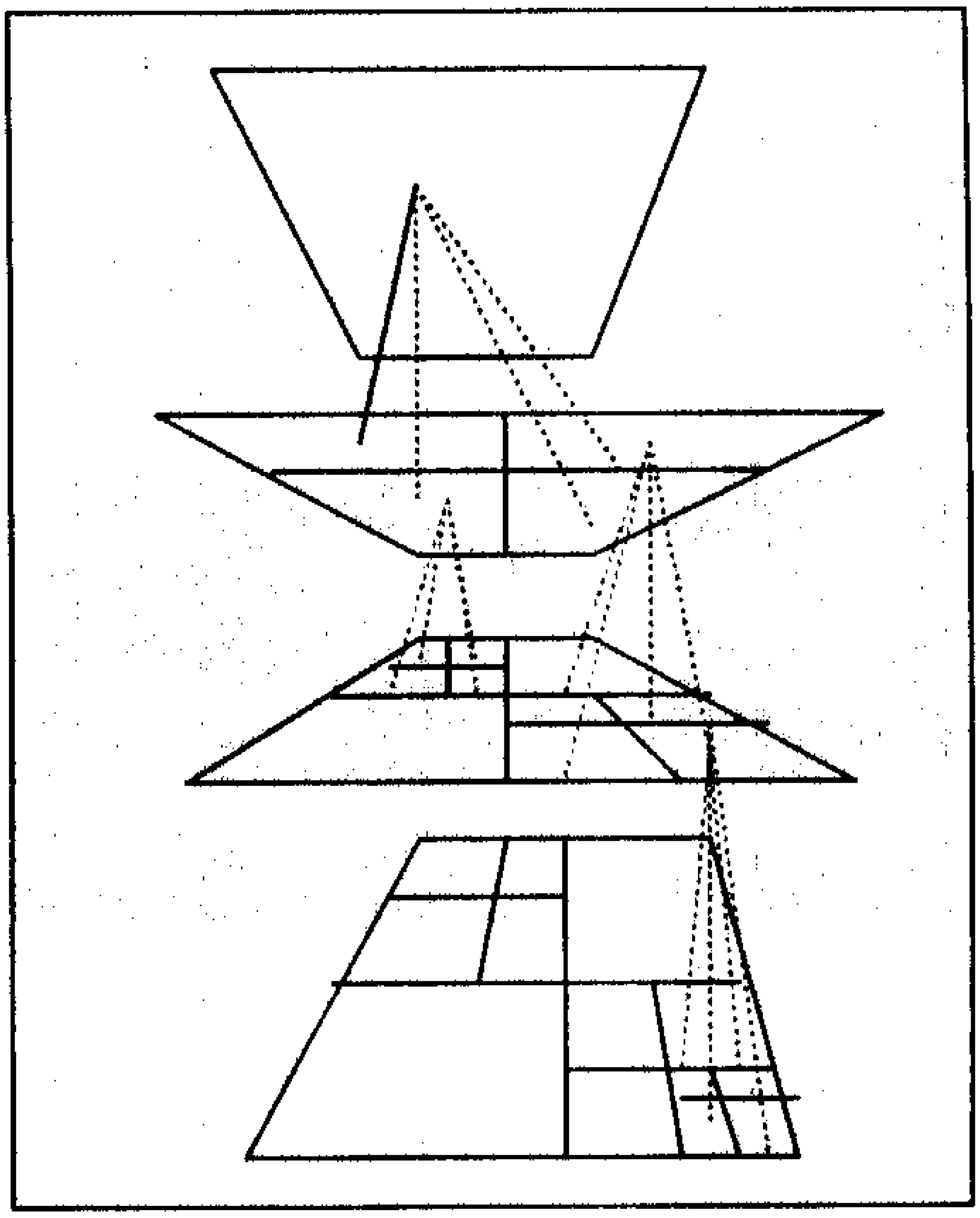


Fig: 4 A Quadtree partition scheme

CHAPTER 3

§3. Quadtree Based Fractal Image compression :

3.1 Encoding :

The ranges are selected as follows : after some initial number of quadtree partitions are made (corresponding to a minimum tree depth) , the squares at the nodes are compared with domains from the domain library i.e. the set of classified domains, which are twice the range size . The pixels in the domain are *averaged in groups of four* so that the *domain* is reduced to the size of *range* , and the affine transformations of pixel values (i.e. scaling and offset) is found that minimises the **rms** difference between the transformed domain pixel values and the range pixel values . All the potential domains are compared with the range . If the resulting optimal rms value is above a pre-selected threshold and the depth of the quadtree is less than a pre-selected maximum depth , then the range square is subdivided into four quadrants , ie add four sub-nodes to the node corresponding the range , and the process is repeated . If the **rms** value is below the threshold , the optimal domain and the affine transformation on the pixel values are stored -- this constitutes one map w_i . The collection of all such maps W = U w_i constitutes the encoding .

But our aim is to minimize the stored bits . for this, first , affine transformation on the pixel values consists of a multiplication by a scaling factor s_i and an addition of an offset value o_i . These values must be quantized for efficient storage , and it is these quantized values that are used when computing the rms difference . Second , if the magnitude of the optimal scaling factor $| s_i |$ is greater than 1 for any of the w_i, there is a chance that the resulting map W will not be eventually contractive . Thus, $| s_i |$ larger than some maximal value s_{max} are truncated.

3.2 The calculation of RMS metric

Suppose there are two image containing n pixel intensities , a_1 , a_2 a_n from domain and b_1 , b_2 , b_n from range , then we want to construct s and o to minimize the quantity

$$R = \sum_{i=1}^{n} (s . a_i + o - b_i)^2$$

This will give us the contrast and brightness setting . The minimum of R occurs , when the partial derivatives with respect to s and o are zero, which occurs when

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

and

$$o = 1/n \left[\sum_{i=1}^{n} b_i - s \sum_{i=1}^{n} a_i \right]$$

In this case ,

$$R = 1/n \left[\sum_{i=1}^{n} b_i 2 - s(s \sum_{i=1}^{n} a_i^2 - 2\sum_{i=1}^{n} b_i a_i + 2o\sum_{i=1}^{n} a_i) + o(n.o - 2\sum_{i=1}^{n} b_i) \right]$$

If denominator of s is zero then $s = 0$ and $o = \frac{1}{n}\sum b_i$
and the rms error is equal to R .

3.3 The Classification :

we use a classification scheme in order to minimize the number of domains compared with a range . Before the encoding , all the domains in the domain library are classified ; this avoids the reclassification of domains . During the encoding of a potential range is classified , and only domains with the same (or near) classification are compared with the range . This significantly reduces the number of domains range comparisons . By near we mean squares that would have been classified differently if their pixel values were slightly different .

There are many classified schemes , we used here the following scheme for less calculation and minimum run time . The scheme we use here is : a square sub-image into upper left , upper right , lower left, and lower right quadrants , numbered sequentially. On each quadrant we compute values proportional to the average and variance : if the pixel values in quadrant *i* are r_1^i , r_2^i , ... r_n^i , *for* i = 1,2,3,4 , we compute

$$A_i = \sum_{j=1}^{n} r_j^i$$

and

n

$$V_i = \sum_{j=1} (r^i_j)^2 - \Lambda_i^{\,2}$$

It is always possible to orient the sub-image so that the Λ_i are ordered in one of the following three ways :

Major class 1 : $\Lambda_1 \geq \Lambda_2 \geq \Lambda_3 \geq \Lambda_4$

Major class 2 : $\Lambda_1 \geq \Lambda_2 \geq \Lambda_4 \geq \Lambda_3$

Major class 3 : $\Lambda_1 \geq \Lambda_4 \geq \Lambda_2 \geq \Lambda_3$

Once the rotation of the square has been fixed , each of the **3** major classes has 24 subclasses consisting of 24 ordering of V_i . There are 4 rotation and 4 trans formation consisting of 8 orientation for each classes i.e. 72 classes in all . If the scaling value s_i is negative , the ordering of the classes above are rearranged . Therefore each domain is classified in two orientations , one orientation for positive s_i and the other for negative s_i .

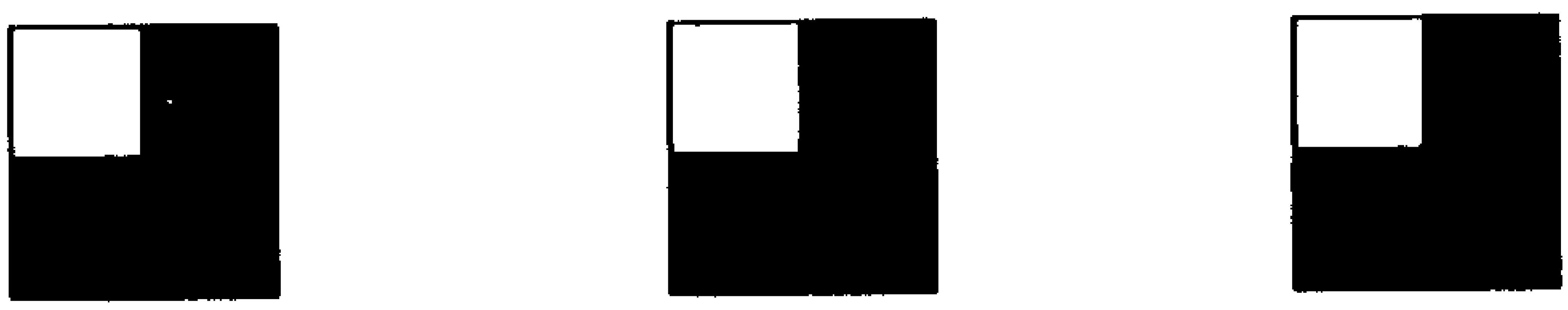


Fig :5 A square image can always be oriented so that the brightness , or average values , of its quadrants fall into one of these canonical positions.

3.4 Decoding :

Decoding an image consists of iterating W from any initial image . The quadtree partition is used to determined all the ranges in the image . For each range R_i , the domain D_i that maps to it is shrunk by two in each dimension by averaging non-overlapping groups of 2 x 2 pixels . The shrunken domain pixels values are then multiplied by s_i , added to o_i, and placed in the location in the range determined by the orientation information . This constitutes one decoding iteration. The decoding step is iterated until the fixed point is approximated ; that is , until further iteration does not change the image or until the change is below some small threshold value .

3.5 Efficient Storage :

To increase the compression , the following bits allocation scheme are used .

Quadtree . One bit is used at each quadtree level to denote a further recursion or ensuing transformation information .

Scaling and Offset . Five bits are used to store the scaling and seven for offset .

Domains . The domains are indexed and referenced by this index .

Orientation . Three bits are used to store the orientation of the domain range mapping .

3.6 Encoding Algorithm :

Input : *The original image and size of the image . also*

i) *The rms tolerance threshold* e_c .

ii) *The maximum depth of the quadtree .partition .*

iii) *The minimum depth of quadtree partition .*

iv) *The maximum allowable scaling factor s_{max}.*

Out put : * *The final quadtree partition of the image .*

- *The scaling and offset values s_i and o_i for each range .*

Step 1: Recursively partition the image and form a Quadtree upto .initial depth plus one.

Step 2 : leave the first some level and take all the nodes from initial depth quadtree as domain and classify them put them in several class pool . All the nodes next to this depth as range , and mark them all uncovered .

Step 3 : For each range find the class in which it belongs to and for each domain in this class find the s_i and o_i , and check whether the root mean square error (rms) .

If (the rms value less than given threshold)

accept this domain, s_i , o_i , and the initial co-ordinate of the domain and the

range and declare this range is covered .

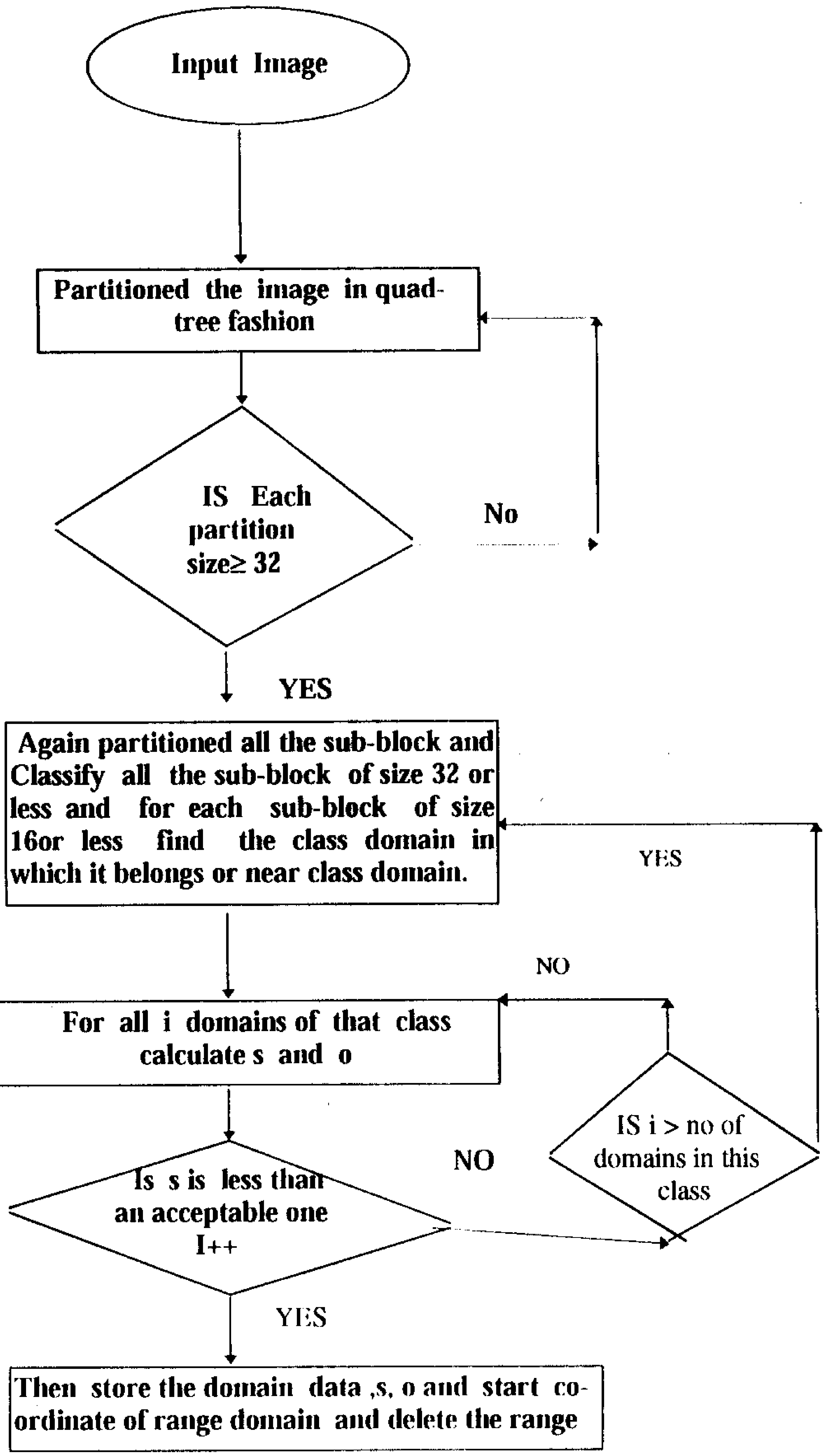
Else

partitioned this range in four square and join them to the four nodes of the

quadtree to this node as a parent of all partition nodes. And add this node to the domain pool .

Step 4 : repeat step 3 until all the range covered or size of the nodes are 4x4.

FLOW CHART FOR ENCODING A IMAGE



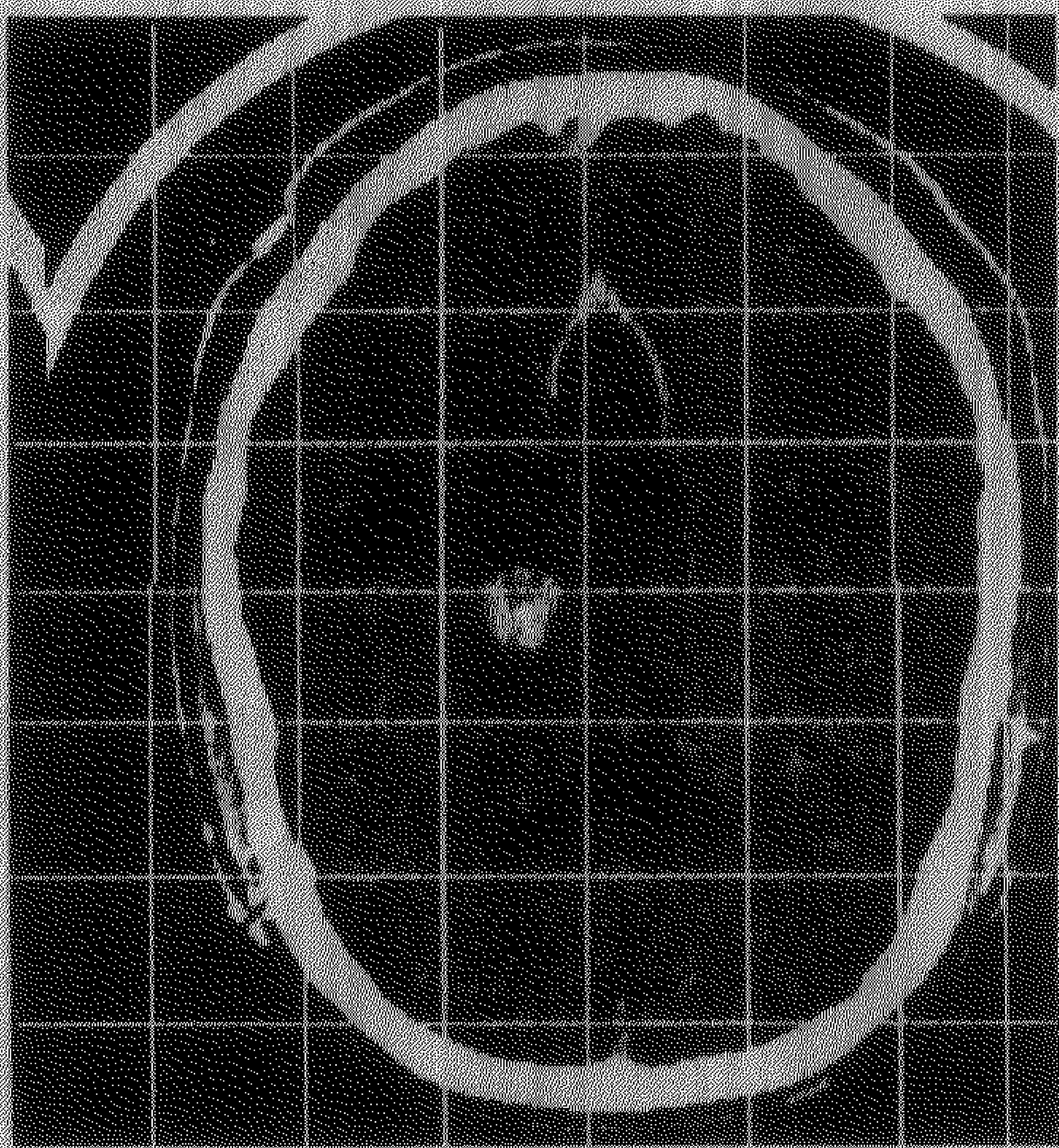


Fig 6Quadtree partitioned scheme

Fig :7 The original MRI image .

Fig 8 This is a decompressed image after three times compressed sequentially.

CHAPTER 4

§4 Wavelet Transformation :

There are many transformations such as Discrete Fourier transformation, Discrete Cosine transformation, Haar-transformation, Walsh-Hadamard transformation etc. We apply here loseless Haar-Walsh-transformation so that there will be no loss after inverse transformation of the image. Consider the following pair of sampling filter for one dimension case $l(x) = f(2x) + f(2x + 1)$ and $h(x) = f(2x) - f(2x + 1)$ which converts a string of length 2n into two strings of length n. One is called a *low pass* filter and the other is called a *high pass* filter . Now the inverse transformation of these two filter is given by $f(2x) = (l(x) + h(x)) /2$ and $f(2x + 1) = (l(x) - h(x))/2$. So in two dimension case the transformation for a image F(x , y) of width 2w and height 2h is given by :

$$LL(x, y) = F(2x, 2y) + F(2x + 1,2y) + F(2x , 2y + 1) + F(2x+1, 2y +1);$$
$$HL(x, y) = F(2x, 2y) + F(2x + 1,2y) - F(2x , 2y + 1) - F(2x+1, 2y +1);$$
$$LH(x, y) = F(2x, 2y) - F(2x + 1,2y) + F(2x , 2y + 1) - F(2x+1, 2y +1);$$
$$HH(x, y) = F(2x, 2y) - F(2x + 1,2y) - F(2x , 2y + 1) + F(2x+1, 2y +1);$$

For the sake of convenience , these four screens are titled into a single screen say P , of dimension 2w x 2h :

$$P(x , y) = LL(x, y) \qquad P (x, y + h) = HL(x , y) ,$$
$$P(x + w, y) = LH(x , y) , P(x +w, y + h) = HH(x, y).$$

The following figures shows this transformation :

Fig :9 Lossless Haar- Walsh - Hadamard Transformation

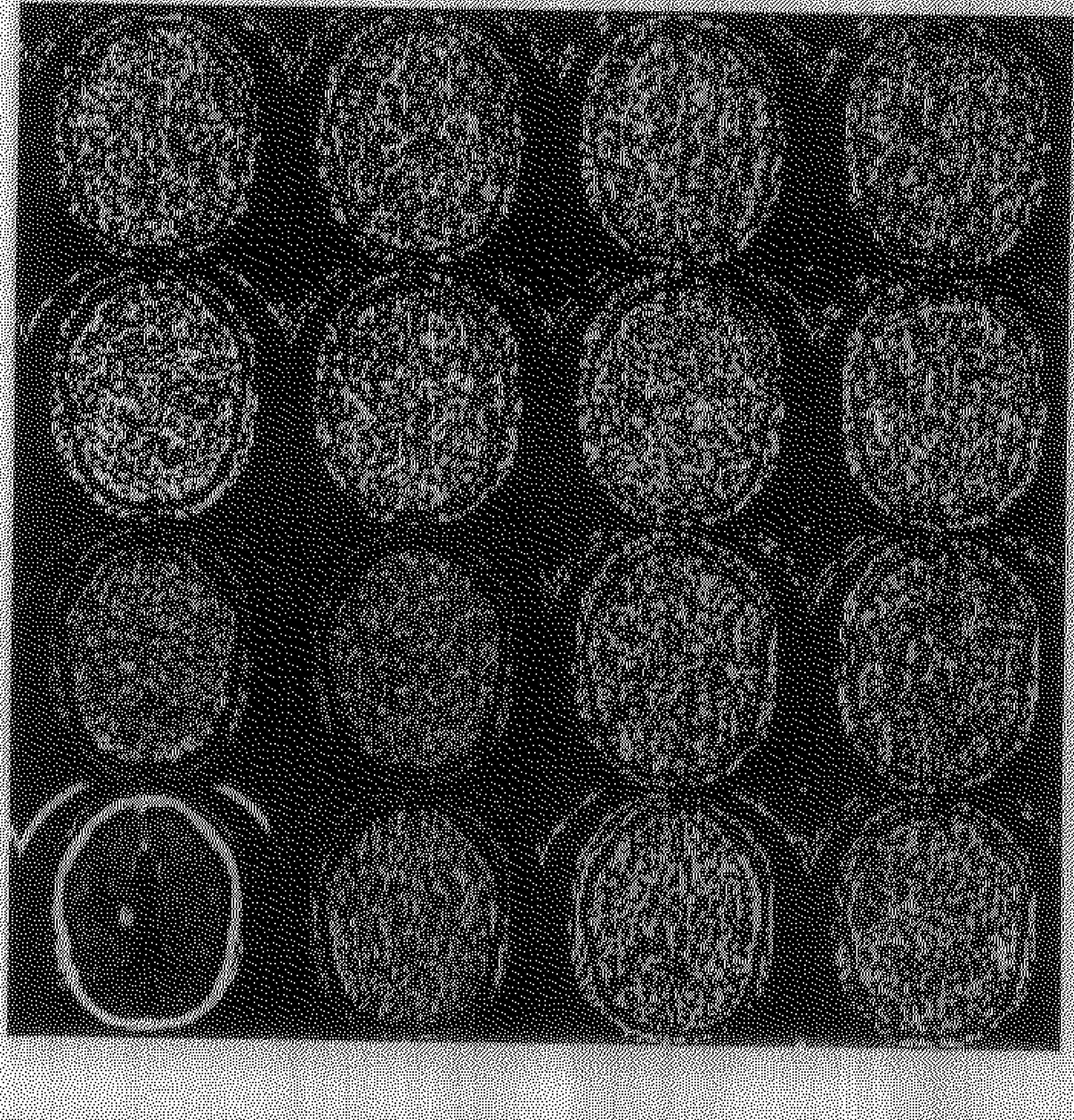


Fig 10applying the same transformation for each quadrant separately.

4.2 Wavelet Compression method :

The wavelet compression scheme compression scheme , has basically three stages

i). *Transformation* *: Which is an invertible or "lossless" transformation which decorelates the mutually dependent parts of the image* ;

ii) Quantization : *Which replace transform coefficients with small integer approximation . In theory all distortion in "lossy" is introduced at this stage .*

iii) Remove redundancy *: This an invertible redundancy remover , or entropy coder ,which rewrites the stream of transform of transform coefficients into more efficient alphabet to asymptotically approach the information-theoretic minimum bit-rate.*

To recover an image from the coded , stored data , we use the inverse of the above methods.

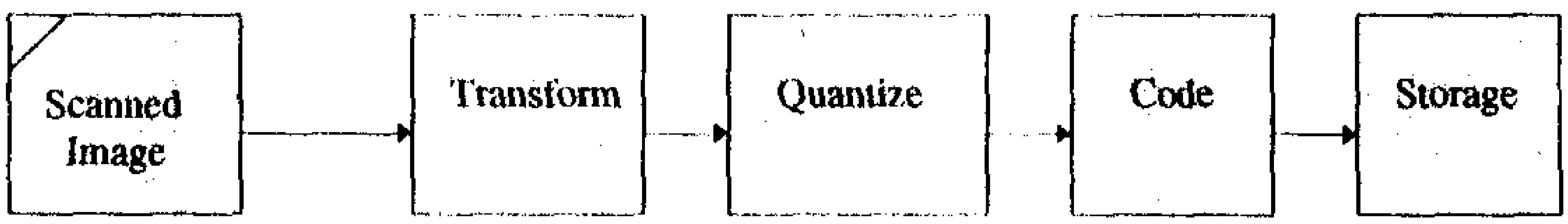


Figure: 1 Transform coding image compression device .

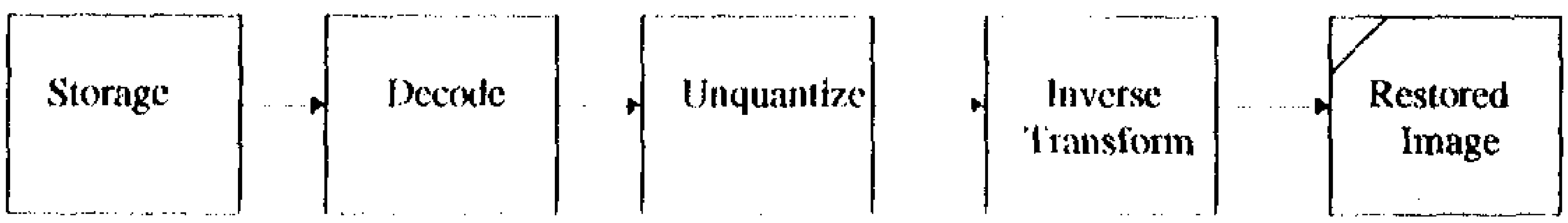


Figure:6 Inverse of transform coder : Decoder

4.3 Fractal Wavelet Compression :

The description of algorithm is very simple , begin with a wavelet transformation of the image and partitioned the transformed image in quadtree partitioned method and then for each four subnode again do the same thing and then
apply fractal compression method for this quadtree and store all the domain after applying quantizer and remove redundancy . The algorithim is given below:

4.3.1 Coding Algorithm :

Input : *the grey level raw image .*

Output : *The coding functions s_i, o_i , quantized parameter , and redundant domain data.*

Step 1: Apply the haar-Walsh-Hadamard transformation to input raw image and store all the four filters low-low , high-low, low-high and high-high to the four subnode of the quadtree.

Step 2 : Repeat step1 until each sub-image size is equal to four by four in bits.

Step 3: Leave the first 3 or 4 depends upon image size which length and breath must be devide by 4 and square type . and take all the nodes in this level as Domain and classify it . and take all the nodes next to that level . as the range . using fractal method try to compressed it.

Step 4 : Now for all domains which have to store find the quantize value and use Hoffinan coding pack it to the fractal code.

4.3.2 Decoding Algorithm:

Input : The coding image

Output: The coding image.

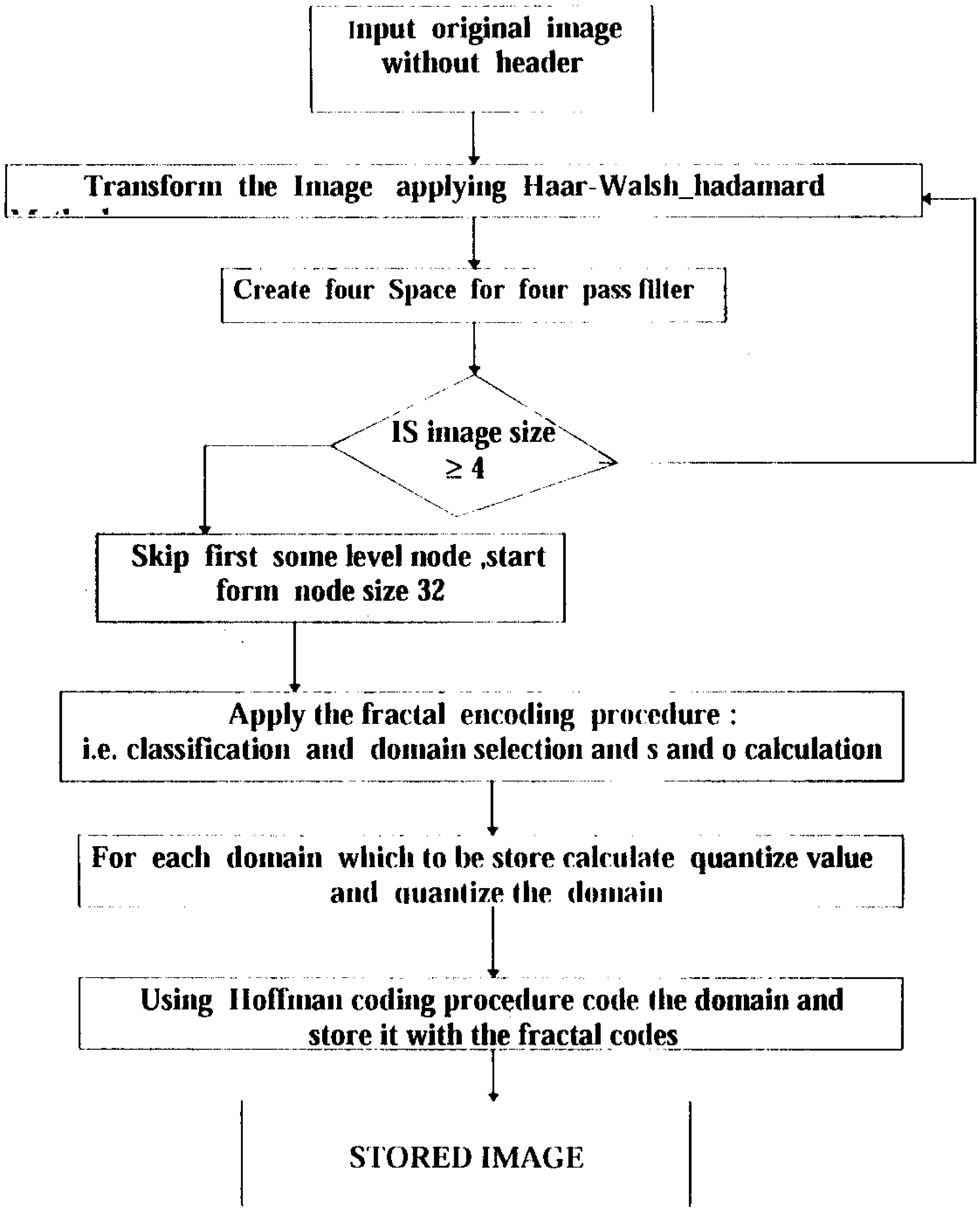
Step 1 : First find the domains and reconstruct the domain applying inverse *Hoffman* coding method and recover the original transformed image .

Step 2: Generate the range using *fractal* decoding method and generate the *quadtree* and generate the complete *quadtree* .

Step 3 : Then from leaf level , using inverse *wavelet* transformation , decode the image .

4.4 Conclusion : There are some works which have to be done to implement this algorithm we already developed the wavelet transformation based quadtree and also the recover method from the leaf level nodes .Recover method is necessary when we want to recover the image we will only generate the leaf level of the quadtree, after adding fractal compression decoding scheme. we also implemeted the method of quantization . Due to insufficient time we can't implement this algorithim completely.

FLOW CHART FOR ENCODING THE FRACTAL - WAVELET MERGINGMETHOD



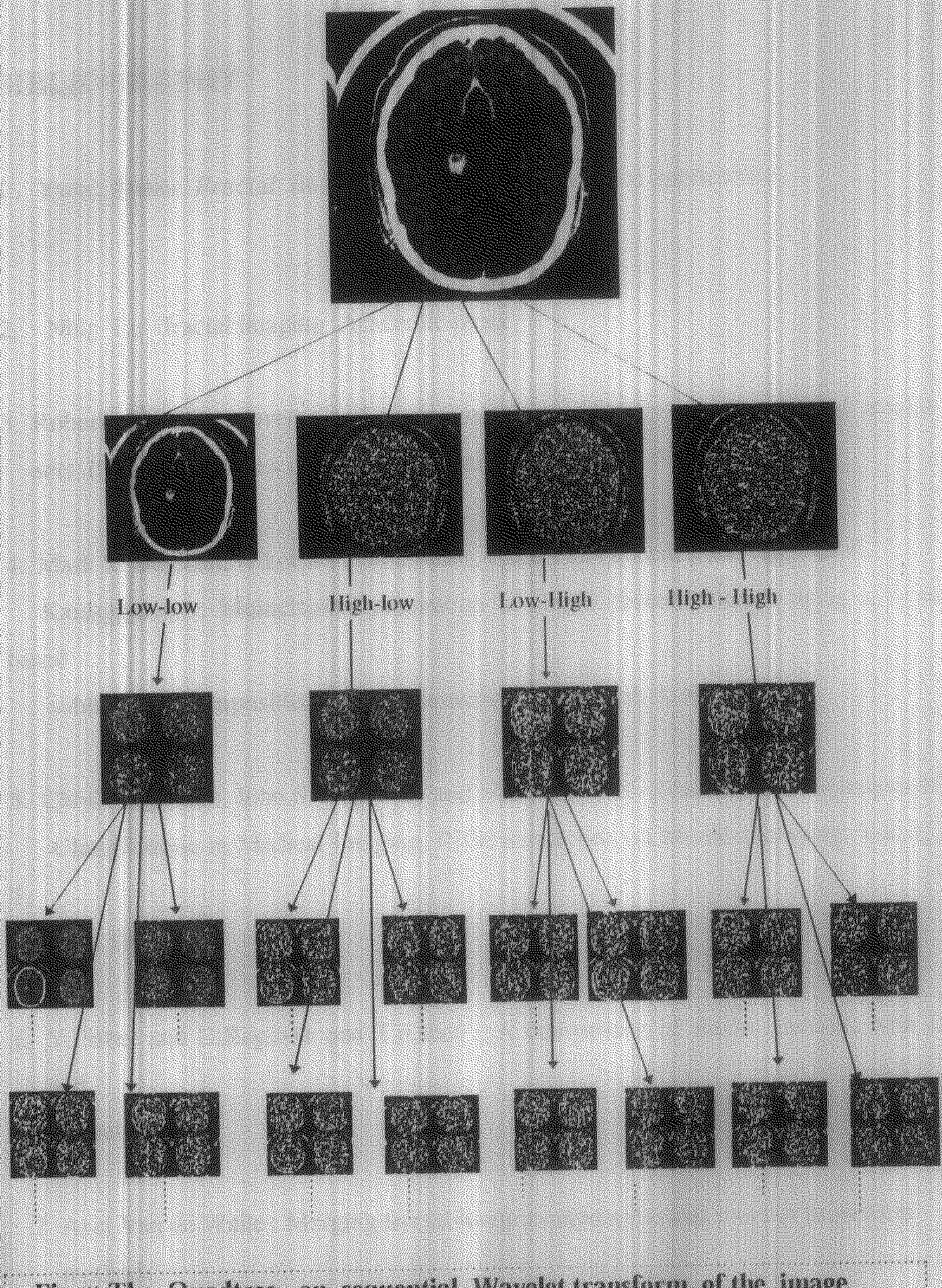


Fig :8 The Quadtree on sequential Wavelet transform of the image
first level is low-low pass , low-high pass , high-low pass and high -high pass filtered 2^{nd} is applying the same on separate portion. and so on .

BIBLIOGRAPHY

1 . Yuval Fisher *Fractal Image Compression Theory and application* .

2. Ning Lu fractal imaging Academic press

3. Roberto Rinaldo : *Inverse and approximation problem for two dimentional fractal sets* .IEEE transactions on image processing, vol. 3, No. 6, Nov. 1994.

4. Mark R. Banham, Nikolas P. Galatsanos, Hector L. Gonzalez and Aggelos K. Katsaggeloos : Multichannel restoration of single channel images using a wavelet-based subband decomposition .IEEE transactions on IP, vol. 3, No. 6, 1994

5. Chi-Chang Chen, John S. Dapoonte and Martin D. Fox : Fractal Feature Analysis and Classification in Medical Imaging, IEEE transaction on Medical imaging, Vol. 1, No. 2, june 1998.

6. Y.Fisher and D.Rogovin and T.p.shen : A comparison of fractal methods with DCT and wavelets.

7 Axel Van de Walle : Merging fractal image compression and wavelet trransform methods.

-------- 0 --------