

Military Technical College  
Kobry Elkobbah,  
Cairo, Egypt



2<sup>nd</sup> International Conference  
on Electrical Engineering  
ICEENG 99

## EVALUATING SHAPE ROUGHNESS USING FRACTAL DIMENSIONS

*M. SHAARAWY IBRAHIM \**

### ABSTRACT

Fractal dimensions provide an objective means for comparing fractals. They are important because they can be derived from real world data, and they can be measured approximately by means of experiments. Also, they may be viewed as measurement of the shape roughness.

In this work, three algorithms for evaluation of fractal dimensions are implemented. These algorithms are based on box counting approach. A comparison is made between the results of these algorithms when evaluating the fractal dimensions of some computer-generated surfaces. These surfaces are of different fractal dimensions. The results showed that the fractal dimension could be used for measuring the shape roughness with an acceptable accuracy. The results showed also that there are significant differences between these methods in accuracy, stability, reliability and the needed time for computation.

### KEYWORDS

Fractal Dimensions, Box Counting, Self-Similarity, Divider Dimension, Shape Roughness.

---

• Associate professor, Dpt. Of Computer & O.R., Military Technical College, Cairo, Egypt.



## 1- INTRODUCTION

The texture of a surface has some properties such as fine, coarse, smooth, granulated, rippled, mottled, irregular, random,, etc. [1]. There are many attempts to find out a model for describing and analyzing textures. The objective of image modeling involves the construction of models for the specification of images. These models can describe images that are observed and they can also be used to generate synthetic images from the model-parameters [2]. Textures in the real world are usually irregular in which structural errors; distortion, or even structural variation are frequent. This means that no strict rule can be used to describe a texture in reality. Another approach to texture description is to use **fractals**, [3], to describe irregular but ordered shapes and surfaces. For example, Normand and Peleg[4], and Barrett and Peleg [5] studied the food structure using the fractal model. Measured fractal dimension also has been related to product functionality, processing history and storage conditions. Olsen *et al.*[6], introduced a new concept called a modified fractal dimension. This modified fractal dimension combines the number of landscape patches, their distribution, and shape into an overall measure of landscape diversity. Cross et al [7], examined the structure of the renal arterial system to decide whether or not it has a fractal structure. They found that, the renal arterial tree has fractal structure, and used it as a measure of abnormalities in this structure. Pentland [8] has suggested a method for image segmentation based on fractal dimension, such that, the image is divided into windows of sizes 4×4 or 8×8 pixels. Then, the fractal dimension is evaluated for each window and a histogram is constructed for the fractal dimension and the number of windows. Finally, a threshold value is chosen to break the histogram into two regions. It is found that this method is better than thresholding on the image intensity. The most popular application of fractals is the generation of natural-scene-like images. It is widely used in computer graphics, flying simulation and computer games [9].

The definition of fractal dimension and its use for modeling natural surfaces is introduced in section 1. Evaluation of the fractal dimension using the box counting method by three different algorithms is described in section 3. Section 4 includes a comparison of the results obtained when implementing the three algorithms for evaluating the fractal dimension of computer generated surfaces. Finally a conclusion is presented.

## 2- FRACTAL DIMENSIONS

In order to clarify what is meant by a dimension, let us consider a one-dimensional curve, in which the (x,y) coordinates are, either explicitly or implicitly, functions of single parameter. For example the points on a circle can be specified using a single parameter  $\theta$ . A two-dimensional surface is one whose coordinates are functions of two parameters. This is the natural notion of *Euclidean dimension*. In general, when we attempt to measure a metric property e.g.: length, area, or volume for any object, we consider a measuring tool of size and then we count the number  $n$  of that tool which is required to cover that object, formula (1).

$$M = n \lambda^D \quad (1)$$

Where,  $M$  is the metric property,  $D$  is the topological dimension of the measuring tool,  $\lambda$  is the size of the measuring instruments, and,  $n$  is the count of that tool which is required to cover that object [10].

Mandlbrot, [3], defines the fractal dimension as the unique fractional power that yields a consistent estimate of the metric property for the fractal object under discussion. It provides the correct adjustment factor for all those details smaller than  $\lambda$ . Also it may be viewed as a measurement of the shape roughness [8].

There are various numbers associated with fractals, which are generally referred to as fractal dimensions. Mathematicians have given many notions of dimension such as topological dimension, Hausdorf dimension, self-similarity dimension, divider dimension, box-counting dimension and more. Our discussion is restricted to three of these dimensions that are special forms of Mandlbrot's fractal dimension: Self-Similarity Dimension, Divider Dimension, and Box-Counting Dimension.

## 2.1- Self-Similarity Dimension

An object is said to be self-similar, if we can divide it into smaller pieces, such that every piece is a small copy of the original object scaled down by a scaling factor. The process of division may be considered as the influence of a similarity transformation on the entire object. This relationship may be formed in the following power law:

$$a = 1/s^D \quad (2)$$

Where  $s$  is the scaling factor and  $a$  is the number of pieces. The exponent  $D=1$  for a line,  $D=2$  for a square, and  $D=3$  for a cube, i.e.  $D$  is equal to the topological dimension for these objects. Strictly speaking, if we have a self-similar object then we can find a relation between the number of pieces  $a$  and the scaling factor  $s$  on the form of equation (2), and hence the self-similarity, fractal dimension is given by the following formula:

$$D = \frac{\text{Log}(a)}{\text{Log}(1/s)} \quad (3)$$

To distinguish it from other types of fractal dimension,  $D_s$  denotes the self-similarity-dimension. That is the self-similarity dimension for a line, a square, a cube coincides with the topological dimension for that object. This means that there is a relationship between self-similarity dimension  $D_s$  and the power  $D$  in the power law (2), which describes the measured length in terms of measuring tool size. The fraction part in  $D_s$  is equal to the power  $D$ . This relation may be expressed mathematically as follows:

$$D_S = 1 + D \quad (4)$$

## 2.2- Divider Dimension

Another type of fractal dimension is called divider or compass dimension. This dimension can be computed for a fractal object using the power law, previously introduced, relating the length, area, or volume for the object under discussion and the size of the measurement tool. The exponent D in this law is used to compute the divider dimension for the formula:

$$D_d = 1 + D. \quad (5)$$

This type of fractal dimension is computed for fractal objects that have no clear self-similarity properties such as coastlines. For example, if  $D=0.36$  for a coastline, then we can say that this coastline has fractal (divider) dimension of about 1.36.

## 2.3- Box Counting Dimension

The previous two types of fractal dimensions are hard to compute from raw data images, unless the object under discussion satisfies some properties. That is to compute self-similarity dimension for an object, it must show clear self-similarity properties. On the other hand, to compute divider dimension for an object, we must be able to use a measurement tool to measure it.

### 2.3.1- Mathematical Definition of Box-Counting Dimension

The box-counting dimension is derived from *Minkovski-Bouligand* definition of dimension [11]. If  $E$  is a bounded set in Euclidean space, then  $E(\varepsilon)$  is the set of all points at a distance  $\varepsilon$  from  $E$ . That is, it is the union of all open balls of radius  $\varepsilon$  centered on  $E$ , it is called Minkovski sausage. The Minkovski-Bouligand dimension is given by:

$$D(E) = \lim_{\varepsilon \rightarrow 0} \frac{\log V(E(\varepsilon))}{\log(\varepsilon)} \quad (6)$$

Where  $V(E(\varepsilon))$  is the volume of the set  $E(\varepsilon)$ .

Practically  $V(E(\varepsilon))$  is difficult to be calculated. Thus, it is preferred to count the number  $\Omega_\varepsilon$  of cubes of side  $\varepsilon$  required to cover  $E$ . It should be noted that, these cubes must have disjoint interior. The union of these cubes is a good approximate for Minkovski sausage. That is, we can replace  $V(E(\varepsilon))$  by  $\varepsilon^3 \Omega_\varepsilon$  in equation (6) which yields:

$$D(E) = \lim_{\varepsilon \rightarrow 0} \frac{\log(\Omega_\varepsilon)}{\log(1/\varepsilon)} \quad (7)$$

This is usually called box-counting dimension. Since formula (6) and formula (7) are mathematically equivalent, the box-counting dimension and Minkovski-Bouligand dimensions are the same. We can use both of them to evaluate the fractal dimension of any fractal bounded set by computing  $V(E(\varepsilon))$  or counting  $\Omega_\varepsilon$  for various values of  $\varepsilon$ . Then, we draw a Log-Log diagram between  $\Omega_\varepsilon$  and  $\varepsilon$ . Finally we use the slope of the fitted line of the represented data as an estimate of the fractal dimension. This formula can also be used to evaluate the fractal dimension of a bounded set if its points are determined by their (x,y,z) coordinates. That is, we can use these formulas to compute the fractal dimension for imaged surfaces, where (x,y) coordinates represent the spatial coordinates of the image, and (z) coordinate represents the image intensity.

### 3- EVALUATING FRACTAL DIMENSION

The problem of applying the fractal model to real image represents a subject of many research efforts. But, there is an important question, that is *How could the imaging process map a fractal surface-shape into an image intensity surface?* Pentland [8] has presented evidence that most natural surfaces are spatially isotropic fractals and that intensity images of these surfaces are also fractals. Pentland's work provides the foundation for the use of fractal features in image analysis.

In this section, we will discuss some methods for evaluating the fractal dimension of an object from its image using box-counting dimension. These methods include Voss method, Keller method, and Sarkar method. Algorithms for these methods are also introduced. These methods relate the rate of growth of the number of boxes needed to cover the volume to the fractal dimension [12]. Box counting estimates the fractal dimension as follows: the image is covered by a sequence of rectangular grids, and a log-log plot is generated which plots the number of grid segments which intersect the image intensity surface versus the size of the grid. A slope value is assigned to the plot and is used for the estimate of the fractal dimension.

#### 3.1- Voss Method

Voss has proposed an approach to estimate the fractal dimension by box counting [13]. This approach is based on the probability, by counting the boxes as follows: Define  $P(m,L)$  to be the probability that there are  $m$  points in a box of side length  $L$ , centered about arbitrary points. That is:

$$\sum_{m=1}^{L^3} P(m,L)=1 \quad (8)$$

Consider that the total number of points in the image is  $M$ , and the image is covered by boxes of side length  $L$ , then the number of boxes with  $m$  points inside each box is given by:

$$\frac{M}{m}P(m, L) \quad (9)$$

That is the total number of boxes with side length  $L$  required to cover the whole image is given by:

$$N(L) = \sum_{m=1}^{L^3} \frac{M}{m} P(m, L) = M \sum_{m=1}^{L^3} \frac{P(m, L)}{m} \quad (10)$$

Notice that,  $M$  is constant for all values of  $L$ . So we may let:

$$N(L) = \sum_{m=1}^{L^3} \frac{P(m, L)}{m} \quad (11)$$

Notice that  $N(L)$  is proportional to  $L^{-D}$ . Hence we can use it to estimate the fractal dimension.

To get an estimate of the fractal dimension using this method we proceed as follows. We center a box of side  $L$  at a pixel  $(x, y, f(x,y))$  in the image of a size  $N \times N$ , and we count the number  $m$  of points which fall within that box. Then, we record that number as  $m(L, x, y)$ . We restrict the process of centering to the pixels having neighbors inside the image. That is, we leave out a strip of width  $(L-1)/2$  on the border of the image. In this case the image is reduced to an image of size  $(M-(L-1)/2) \times (N-(L-1)/2)$ . The reduced image is divided into windows. For each window, the occurrences of  $m(L, x, y)$  are accumulated over the pixels within the window, and the probability distribution  $P(m, L)$  is obtained by dividing the occurrence  $m(L, x, y)$  by the total number of pixels in the window. The estimate of the fractal dimension is the slope of the fitted line obtained from applying the least square linear fit method to  $\log(L)$  versus  $\log(N(L))$ . Finally, we calculate  $N(L)$  from equation (11)

### 3.1.1- Algorithm for Voss method

**Begin**

Set  $P[m,L] \leftarrow 0$  For all values of  $L$  and possible values of  $m$ ;

**Do For each pixel in the image**

**Do For each box of length  $L$**

Center a box of side length  $L$  on the current pixel  $(x,y,f(x,y))$ ;

$m \leftarrow$  The number of image points which fall in the box;

Increment  $P[m,L]$  by 1;

**End Do**

**End Do**

```

Do For each value of L and m
    Dividing P[m,L] by the total number of points which falls on that box;
End Do
Do For each value of L and m
    Calculate N[L] from equation (11);
End Do
Estimate D from the slope of the least square linear fit of the data Log(L) and
Log(N[L]);
END.

```

The disadvantage of this method is that, if there are some points on the image intensity surface that are widely spaced in the Z-direction, then the fractal dimension calculated using this method will not be affected by this points. So, the estimated fractal dimension will be less than the appropriate one. One of the main reasons for this, is the quantization effect. Since the original surface is continuous, and when the fractal dimension of that surface approaches 3.0, it becomes very irregular, and after quantization process, the image will contain points with wide differences on the gray level. Hence, counting boxes using Voss's method will yield a large number of boxes having only one point on it; which is the center point. Thus, decreasing the estimated fractal dimension.

### 3.2- Keller Method

To avoid the disadvantage of Voss approach, Keller *et al.* [13] introduced an algorithm for box counting method based, on probability, which is a modification of Voss's algorithm. In this method, the fractal surface between the center point of the box and each of its neighborhoods is approximated by linear interpolation. The newly interpolated surface is intersected with the box and we record the number of points that fall in it. Keller used the following procedure for the process of interpolation. First, for each y, every point (x, y, f(x,y)) on the surface is connected to the points (x-1, y, f(x-1, y)) and (x+1, y, f(x+1, y)) with digital 3D-line. After these lines are generated for all y, the process is repeated holding the x-coordinate fixed. It is important to note that Keller's method is slower than Voss method, due to the more time needed for computation. On the other hand, Keller's method gives satisfactory results, except for images with intensity surfaces with very high fractal dimensions.

### 3.2.1- Algorithm For Keller Method

```

Begin
    Set P[m,L] ← 0 For all values of L and possible values of m;
    Do For each box of length L
        Do For each pixel in the image
            Center a box of side length L on the current pixel (x,y,f(x,y));
            m (x,y,L) ← The number of image points which fall in the box;
            Increment P[m,L] by 1;
        End Do
    End Do

```

Interpolate the surface between the center of the box and each of its neighbors;

Increase  $m(x,y,L)$  by The number of interpolated surface points which fall in the box;

**End Do**

**Do For** each value of  $L$  and  $m$

Evaluate the probability that there are  $m$  points in a box of side length  $L$  by dividing  $P[m,L]$  by the total number of points which falls on that box (the summation of all values of  $m$ );

**End Do**

**Do For** each value of  $L$  and  $m$

Calculate  $N[L]$  from equation (11);

**End Do**

Estimate  $D$  from the slope of the least square linear fit of the data  $\log(L)$  and  $\log(N[L])$ ;

**END.**

### 3.3- Sarkar Method

Sarkar and Chaudhuri [14] have introduced a modification to Keller's method described above. In their modification, the number of boxes required to cover the 3D space is counted in different manner. Consider that an image of size  $M \times M$  pixels has been scaled down to  $S \times S$ , where  $1 < S \leq M/2$ , and  $S$  is an integer. The scaling down ratio is  $S/M$ . As in the previous method, suppose that the image is a 3D space with  $(x,y)$  coordinates denoting 2D position, and the  $z$ -coordinate denotes gray levels. The  $XY$  space is partitioned into grids of sides  $S \times S$ . On each grid there is a stack of boxes of sides  $S \times S \times S'$ . If the total number of gray levels is  $G$  then  $\lfloor G/S' \rfloor = \lfloor M/S \rfloor$ . Let the maximum and minimum gray levels in the  $(i,j)^{th}$  grid fall into boxes number  $K$  and  $L$  respectively. Define

$$n_r(i,j) = L + K - 1 \quad (12)$$

to be the contribution of  $N_r$  (the number of boxes for the scaling down ratio  $r$ ) in the  $(i,j)^{th}$  grid. Taking the sum of the contributions of all grids, we get

$$N_r = \sum_{i,j} n_r(i,j) \quad (13)$$

$N_r$  is computed for various values for  $r$ . Then, we estimate the fractal dimension from the least square linear fitting of  $\log(N_r)$  versus  $\log(1/r)$ . It is important to note that, this method of computing considers the effect of the sharp gray level variation, in neighboring pixels in the image. Hence, it can be used for an accurate estimate for the fractal dimension for rough textures.

### 3.3.1- Algorithm for Sarkar method

```

Begin
  Set  $N[S] \leftarrow 0$  For all values of  $S$ ;
  Do For each pixel in the image
    Do For each grid length  $S$ 
      Center a stack of boxes of side length  $S$  on the current pixel  $(x,y,f(x,y))$ ;
       $K \leftarrow$  the box containing the minimum gray level on the grid;
       $L \leftarrow$  the box containing the maximum gray level on the grid;
      Calculate  $n_r$  from equation (13)
       $sn \leftarrow sn + n_r$ 
    End Do
     $N[S] \leftarrow sn$ 
  End Do
  Estimate  $D$  from the slope of the least square linear fit of the data  $\text{Log}(S)$  and  $\text{Log}(N[S])$ ;
END.

```

Note that, in the previous methods the image is considered as a 3D space as the following form :  $z = f(x,y)$ , with  $x$  and  $y$ -coordinates are supposed to be independent (spatial coordinates), but in most applications  $z$ -coordinate is a measurement whose scale is usually given arbitrarily by a machine. That is, the three coordinates are different in nature. Strictly speaking, the function  $f$  and  $c.f$  (here  $c$  is constant) should define a surface with the same fractal dimension. Unfortunately, algorithms based on the definition of box-counting dimension are sensitive to the change of amplitude [11].

## 4- IMPLEMENTATION OF FRACTAL DIMENSION EVALUATION ALGORITHMS

In order to use the above three methods for evaluating the fractal dimensions, their algorithms are implemented and tested with a set of computer generated surfaces with different fractal dimensions. The use of computer generated surfaces comes from the need to test these algorithms for evaluating the fractal dimension for surfaces of predefined fractal dimensions. The fractional Brownian motion (fBm) is used to generate the fractal surfaces. The generated graphics shapes are converted to image files in order to be suitable for processing.

### 4.1- Generation of Surfaces

#### 4.1.1- Brownian Motion

Brownian motion is the motion of small particles caused by the continual bombardment by other neighboring particles [15]. The fractional Brownian motion is a generalization of ordinary Brownian motion that has been used successfully to model a variety of natural phenomena, such as terrain, and clouds.

#### 4.1.2: Ordinary Brownian Motion

Brownian motion is a process in which a particle is subjected to random displacement. In a simple one-dimensional model the displacement are either +1 struck from the left or -1 struck from the right. More formally, the position of the particle at time  $t$  can be represented as a random variable  $V(t)$ . That is,  $V(t)$  can be written as the following equation:

$$V(t) = \sum_{i=0}^t L_i \quad (14)$$

where  $L_i$  is a random variable taking the values +1 and -1 with equal probability. One characteristic of such a random walk is that the average displacement is proportional to the square root of the time difference, that is:

$$|V(t) - V(t + dt)| \propto dt^{0.5} \quad (15)$$

This is due to the huge number of particles moving in virtually all directions, so the average position remains the same [13], [29].

#### 4.1.2- Fractional Brownian Motion

Fractional Brownian motion (fBm) is a generalization of ordinary Brownian motion with the scaling property:

$$|V(t) - V(t + dt)| \propto dt^h \quad (16)$$

for arbitrary  $h$ ,  $0 < h < 1$ . The exponent  $h$  is called *Hurst exponent*. Ordinary Brownian motion has  $h=1/2$ . The concept of fractional Brownian motion can be extended to higher dimensions when the variable  $t$  becomes a vector. If  $t$  is a two-dimensional vector then  $V(t)$  represents a surface in the space.

#### 4.2- Generating Fractional Brownian Motion

Meader [16], developed three different methods for Generating two-dimensional fractional Brownian motion. It is important to note that, the two-dimensional fractional Brownian motion is represented by a surface. These methods are namely *Random Addition*, *Fourier synthesis*, and *Random Faults*. In this work we use the second method (Fourier synthesis) to generate these surfaces because it is simple in calculations.

##### 4.2.1- Fourier Synthesis

The average amplitude  $v_f$  of fBm data at frequency  $f$  obeys the simple law

$$v_f \propto 1/f^{\beta/2} \quad (17)$$

where  $\beta = 2h+1$  is the spectral exponent. One method to generate fBm data is to generate random Fourier coefficient according to that formula and then to perform inverse Fourier transforms.

In two-dimensional case, the Fourier coefficient  $v_{k_1 k_2} = v_k$  should have mean amplitude

$$|K|^{-(\beta+1)/2} = (k_1^2 + k_2^2)^{-(\beta+1)/4} = (k_1^2 + k_2^2)^{-(h+1)/2} \quad (18)$$

the fractal dimension of the generated surface is related to Hurst exponent by the following formula:

$$D = E+1-h \quad (19)$$

### 4.3- Results

#### 4.3.1- Effect Of Image Resolution

The results are shown in fig.(1), fig(2) and fig(3). We noticed that, these methods are affected by the image resolution. Upon increasing the image resolution from 32×32, 64×64 to 128×128, the estimated fractal dimensions get more accurate values.

Box-counting limitations come from the assumed limits on the range of scale-invariance and on methods of assigning slopes to non-linear plots. It leads to unreliable and inconsistent FD estimates for low-resolution images [16].

The estimated fractal dimensions using methods of Voss and Keller show-in general- an unstable behavior for images with size 32×32.

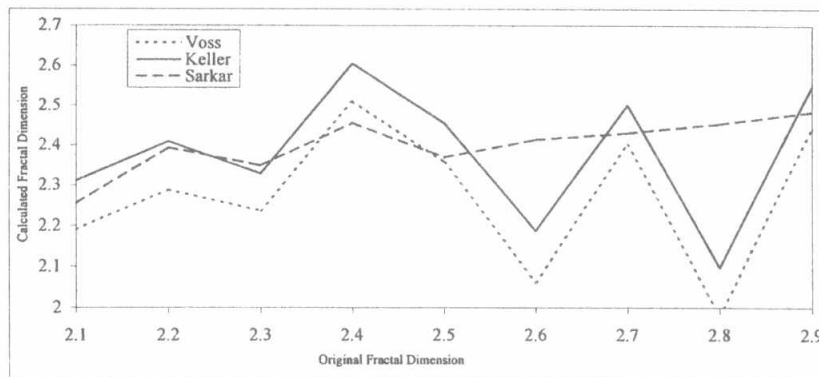


Fig (1): fractal dimension for generated surface of size 32×32 by different methods

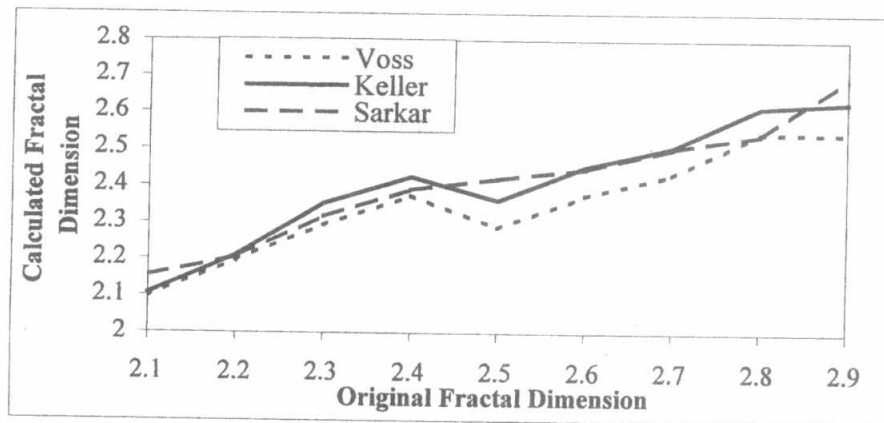


Fig (2): fractal dimension for generated surface of size 64×64 by different methods

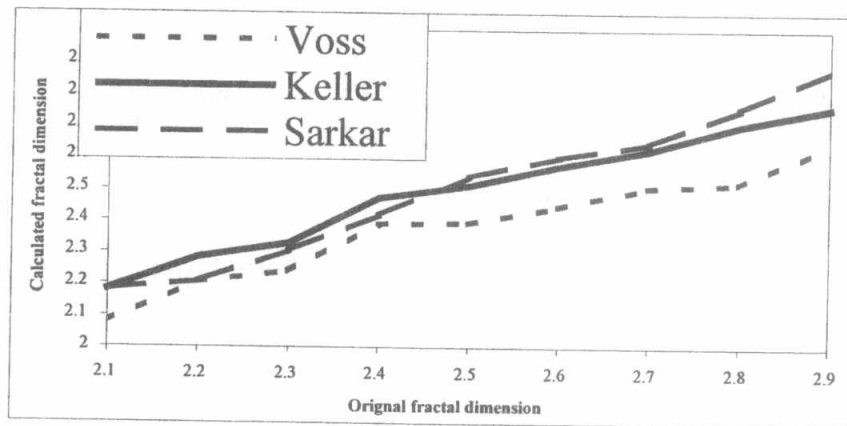


Fig (3): fractal dimension for generated surface of size 128×128 by different methods

At resolution 64×64, acceptable results are achieved. But, for surfaces with original fractal dimension 2.5 there is a drop in results obtained by Voss and Keller methods.

On images with resolution 128×128, as expected, the estimated fractal dimensions lie in the range 2.0-3.0, and increases with the increment of image roughness. Note that, Voss's method compresses the estimated fractal dimension to the middle of the true range as Keller stated in his paper [13]. The interpolation in Keller's method gives satisfactory results up to certain level of roughness of the image intensity surface.

Sarkar's method gives more accurate results than the previous two methods. An example for this accuracy, we noticed that the estimated fractal dimensions from Voss and Keller methods decrease with the increasing of surface roughness for some surfaces (different images). On the other hand, Sarkar's method, always, increases with the increasing of the surface roughness.

In general, the results obtained using Sarkar's method are better than those obtained using the other two methods, because it is accurate and more sensitive for surface roughness. It almost covers the total range of fractal dimension.

#### 4.3.2- Computation Requirements

A comparison is made for the number of computational operations and the elapsed time while using the three methods. Table (3) and table (4) depict this comparison. The results showed that Sarkar's method represents better performance than the other two methods.

Table (3): Comparison of number of computations

Method	Operations				
	Add	Subtract	Multiply	Divide	Compare
Voss	413570	101736	20	4072	312283
Keller	687633	237176	20	43408	400094
Sarkar	290644	3439	20	40	560654

Table (4): Comparison of elapsed time

Method	Elapsed time for different Resolution		
	32x32	64x64	128x128
Voss	0.001	0.050	0.22
Keller	0.001	0.065	0.310
Sarkar	0.001	0.059	0.110

## CONCLUSIONS

In this work we discussed the fractal model, which is one of the most important image models. We adopted a number of the most popular algorithms for evaluating the fractal dimension from digital image data. The advantages and disadvantages of each algorithm are, also, illustrated. These results showed that Sarkar's method more reliable and accurate than Voss and Keller methods for evaluating the fractal dimension. Sarkar's method shows, also, better performance in computational requirements. Finally, the fractal dimension represents suitable measure of surface roughness natural surfaces, clouds, terrain, etc..

## REFERENCES

- [1] Ahuja, N., A Rosenfeld, "Mosaic Models for Textures", IEEE Trans. Pattern Analysis and Machine Intelligence, Vol. 3, No. 1, 1981.

- [2] Cross, G. R., Anil K. J., "Markov Random Field Texture Models", IEEE Trans. Pattern Analysis and Machine Intelligence, Vol. 5, No. 1, 1983.
- [3] Mandlbrot, B. B., "The Fractal Geometry of Nature", Freeman, San Francisco, CA, 1983.
- [4] Normand, M. D., M. Peleg, "Determination of Fractal Dimension of Particle Silhouette Using Image Processing Techniques", Powder Technology, Vol. 45, No. 3, pp. 271-275, 1986.
- [5] Barrett, A. H., M. Peleg, "Application of Fractal Analysis to Food Structure", Lebensm.-Wiss. U. Technol., Vol. 28, No. 6, pp. 553-563, 1995.
- [6] Olsen E., R. D. Ramsey, D. S. Winn, "A modified Fractal Dimension as a Measure of Landscape Diversity", Internet at: [www.us.net/aspes/issues/jul94/art1b.html](http://www.us.net/aspes/issues/jul94/art1b.html), 1995.
- [7] Cross, S. S., R. D. Start, P. B. Silcocks, A. D. Bull, D. W. Cotton, J. C. Underwood, "Quantitation of The Renal Arterial Tree by Fractal Analysis", J. Pathology, Vol. 170, pp. 479-484, 1993.
- [8] Pentland, A. P., "Fractal-Based Description of Natural Scenes", IEEE Transaction on Pattern Analysis and Machine Intelligence, Vol. 6, No. 6, 1984.
- [9] Wenger, T., B. Tyler, "Fractal Creations", Wait Group Press, 1993.
- [10] Barnsley, M., "Fractals Every Where", Academic Press Inc., \*\*
- [11] Mandlbrot, B. B., "The Fractal Geometry of Nature", Freeman, San Francisco, CA, 1983.
- [12] Ohanion, P. P., R. C. Dubes, "Performance Evaluation for Four Classes of Texture Feature", Pattern Recognition, Vol. 25, No. 8, pp. 819-833, 1992.
- [13] Keller, J. M., S. Chen, R. M. Crownover, "Texture Description and segmentation Through Fractal Geometry", Computer Vision, Graphics and Image Processing, Vol. 45, pp. 150-166, 1989.
- [14] Sarkar, N., B. B. Chaudhuri, "An Efficient Approach to Estimate Fractal Dimension of Textural Images", Pattern Recognition, Vol. 25, No. 9, pp. 1035-1042, 1992.
- [15] Finlay, M., K. A. Blanton, "Real-World Fractals", M&T-Books, 1993.
- [16] Meader " Mathematica Software Package".

