more precise measurements should yield a larger rather than a smaller speed ratio.

V. Conclusion

The fringe distance measure is a simple, intuitively appealing measure of the similarity of two binary images. It can be calculated rapidly and easily, and does not have any parameters to be optimized. It provides a degree of distortion and displacement tolerance without loss of resolution. It seems to perform about as well for template matching as use of Gaussian blurred images with optimized blurring parameter.

In many applications, the increased speed of this distance measure would allow a larger number of templates to be used, which would increase recognition accuracy. The fringe distance measure was developed as part of a contract with the U.S. Postal Service to develop high-speed equipment for reading machine printed addresses on envelopes. The large number of fonts encountered in this application requires a very large number of templates. The design of custom hardware to accelerate the computation was greatly aided by the fact that it uses no arithmetic other than the addition of small integers.

References

[1] R. D. Brandt, Y. Wang, A. J. Laub, and S. K. Mitra, "The recognition of shapes in binary images using a gradient classifier," *IEEE Trans. Syst., Man, Cybern.*, vol. 19, pp. 1595–1599, Nov./Dec. 1989.

[2] D. H. Ballard and C. M. Brown, *Computer Vision*. Englewood Cliffs, NJ: Prentice-Hall, 1982.

[3] D. Marr and T. Poggio, "A computational theory of human stereo vision," *Proc. Roy. Soc. Lond.*, vol. B204, pp. 301–328, 1979.

[4] D. Marr and E. Hildreth, "Theory of edge detection," *Proc. Roy. Soc. Lond.*, vol. B207, pp. 187–217, 1980.

[5] D. Marr, *Vision: A Computational Investigation into the Human Representation and Processing of Information*. New York: Freeman, 1982.

[6] A. L. Yuille and T. A. Poggio, "Scaling theorems for zero-crossings," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 8, pp. 15–25, Jan. 1988.

[7] J. Babaud, A. P. Witkin, M. Baudin, and R. O. Duda, "Uniqueness of the Gaussian kernal for scale-space filtering," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 8, no. 1, pp. 26–33, Jan. 1988.

[8] A. Rosenfeld and A. C. Kak, *Digital Picture Processing*. New York: Academic, 1976, pp. 352–353.

[9] T. Ohtsuki, *Layout Design and Verification*. New York: North-Holland, 1986, pp. 99–113.

An Efficient Differential Box-Counting Approach to Compute Fractal Dimension of Image

Nirupam Sarkar and B. B. Chaudhuri

***Abstract*—Fractal dimension is an interesting feature proposed recently to characterize roughness and self-similiarity in a picture. This feature has been used in texture segmentation and classification, shape analysis and other problems. An efficient differential box-counting approach to estimate fractal dimension is proposed in this note. By comparison with four other methods, it has been shown that our method is both efficient and accurate. Practical results on artificial and natural textured images are presented.**

Manuscript received October 17, 1991; revised June 2, 1992 and March 5, 1993.

The authors are with the Electronics and Communication Sciences Unit, Indian Statistical Institute, Calcutta—700 035, India.

IEEE Log Number 9212941.

I. Introduction

Simple objects can be described by the ideal shape primitives, such as cubes, cones and cylinders. But most of the natural objects are so complex and erratic that they cannot be described in terms of simple primitives. On the other hand, the concept of self-similarity seems to play an important role in the description of nature. The complex and erratic shape description in terms of self-similarity was introduced by Mandelbrot [1], who proposed the fractal geometry of nature.

The concept of fractal dimension (FD) can be useful in the measurement, analysis, and classification of shape and texture. Pentland [2], [3] noticed that the fractal model of imaged three-dimensional (3-D) surfaces can be used to obtain shape information and to distinguish between smooth and rough textured regions. Rigaut [4] used the concept for image segmentation. Some of the other applications involve sedimentology and particle morphology [5], [6], image data compression [7], [8] and computer graphics [9].

Several approaches exist to estimate the FD in image. For example, Peleg [10] used the ϵ-blanket method, which is a 2-D generalization of the original approach suggested by Mandelbrot [1]. Pentland [2] considered the image intensity surface as fractal Brownian function (fBf) and estimated FD from Fourier power spectrum of fBf. Gangepain and Roques-Carmes [11], as well as Keller *et al.* [12], used variations of box-counting approach to estimate FD.

It is useful to compare the approaches and suggest, if possible, an improved one that is computationally attractive and gives accurate results. Our correspondence is motivated to this end. The basics of FD and different approaches of its estimation are revised in Section II. In Section III a differential box-counting approach is proposed. The modified approach is compared with the other approaches in terms of computer complexity and accuracy.

II. Basic Definition and Estimation Approaches

A set is called a fractal set if its Hausdorff-Besicovitch dimension is strictly greater than its topological dimension. Mandelbrot [1] coined the term fractal from the Latin word *fractus*, which means irregular segments.

Mandelbrot first described an approach to calculate FD while estimating length of coastline. Consider all points with distances to the coastline of no more than ϵ. These points form a strip of width 2ϵ, and the suggested length $L(\epsilon)$ of the coastline is the area of the strip divided by 2ϵ. As ϵ decreases $L(\epsilon)$ increases. Mandelbrot studied that for many coastlines the following formula holds well:

$$L(\epsilon) = F\epsilon^{1-D} \tag{1}$$

where F and D are constants for a specific coastline. He called D the fractal dimension (FD) of the line. D can be derived from least square linear fit of a log–log plot of $L(\epsilon)$ and ϵ. If m is the slope of the fitted line, then the FD of curve (coastline) will be $1 - m$. Note that m is always negative.

Peleg *et al.* [10] adopted Mandelbrot's idea and extended it to surface area calculation. In this extension, the image can be viewed

as a hilly terrain surface whose height from the normal ground is proportional to the image gray value. Then all points at distance ϵ from the surface on both sides create a blanket of thickness 2ϵ. The estimated surface area is the volume of the blanket divided by 2ϵ. For different ϵ, the blanket area can be iteratively estimated as follows: The covering blanket is defined by its upper surface, u_ϵ and the lower surface b_ϵ. Initially, given the gray level function $g(i, j)$, $u_o(i, j) = b_o(i, j) = g(i, j)$. For $\epsilon = 1, 2, 3, \cdots$, blanket surfaces are defined as follows:

$$u_\epsilon(i, j) = \max \{u_{\epsilon-1}(i, j) + 1, \max_{d(i,j,m,n) \le 1} u_{\epsilon-1}(m, n)\}$$

$$b_\epsilon(i, j) = \min \{b_{\epsilon-1}(i, j) + 1, \min_{d(i,j,m,n) \le 1} b_{\epsilon-1}(m, n)\}$$

where $d(i, j, m, n)$ is the distance between pixels $(i, j$ and $m, n)$. The volume of the blanket is given by

$$v_\epsilon = \sum_{i,j} (u_\epsilon(i, j) - b_\epsilon(i, j))$$

while the surface area is measured as

$$A(\epsilon) = \frac{(v_e - v_{\epsilon-1})}{2}.$$

The area of fractal surface behaves according to the expression

$$A(\epsilon) = F\epsilon^{2-D}. \tag{2}$$

Fractal dimension can be derived from least square linear fit of the log–log plot of $A(\epsilon)$ and ϵ, with the help of (2).

The recent results in the field of fractal model and the fractal geometry point out that most of the natural objects are not ideal but semi-fractal. As a result, often two linear regions appear in the plot as two fractional elements [13]. The first element characterizes the edge or surface effects of the analysed image and is called *textural fractal dimension.* The second element describes the object at a higher resolution level and is called *structural fractal dimension.* These two quanties can be defined and calculated in the same way, applying the same algorithm but for different image resolution.

Pentland [2] suggested a method of estimating FD by using the Fourier power spectrum of image intensity surface. It can be shown that the Fourier power spectrum $P(f)$ of fractal Brownian function (f) is proportional to f^{-2h-1}, where $h = 2 - D$, and D is the FD. From least square fit of the log–log of $P(f)$ and f, one can estimate FD of an image intensity surface, provided image intensity surface can be modeled as a fractal Brownian function.

Mandelbrot stated that one criterion of a surface being fractal is its self-similarity. Self-similarity can be explained as follows. Consider a bounded set A in Euclidean n-space. The set is said to be self-similar when A is the union of N_r distinct (nonoverlapping) copies of itself each of which is similar to A scaled down by a ratio r. Fractal dimension D of A can be derived from the relation [1]

$$1 = N_r r^D \text{ or } D = \frac{\log (N_r)}{\log (1/r)}. \tag{3}$$

However, natural scenes practically do not exhibit deterministic self-similarity. Instead, they exhibit some statistical self-similarity. Thus, if a scene is scaled down by a ratio r in all n dimensions, then it becomes statistically identical to the original one, so that (3) is satisfied.

It is difficult to compute D using (3) directly. An approximate method used by Gangepain and Roques-Carmes [11], called the *reticular cell counting* approach is as follows. Consider the 3-D space where two coordinates (x, y) represent 2-D position and the third (z) coordinate represents the image intensity. For a given scale L, partition the 3-D space into boxes of sides $L \times L \times L'$, where L can be a multiple of the sidelength of a pixel in (x, y) and L' can be a multiple of the gray level unit in z-direction. If G is total gray levels and $M \times M$ is the size of image, then $L' = \lfloor L \times G/M \rfloor$. Let, for $L = 1$, the box be called *space-intensity cell* or *spicel.* Then, for $L = 3$, the box contains $3 \times 3 \times 3 = 27$ spicels. Suppose we can cover the 3-D space by a 3-D box of size $L_{\max}$. Then $L = r \times L_{\max}$. Changing parameter from r to L we have, from (3),

$$N_L = \frac{1}{r^D} = \left[\frac{L_{\max}}{L}\right]^D$$

i.e.,

$$N_L \propto L^{-D}. \tag{4}$$

If a box contains at least one sample of gray level intensity surface, we can call it a countable box. For a fixed L, let the total number of countable boxes be N_L. Several values of the L are chosen and the least square linear fit of $\log N_L$ versus $\log L$ gives the value of $-D$. But when the actual FD of an image is very high, points on the image intensity surface become widely spaced in the z-direction, effectively lowering the estimated FD. From Fig. 3 it is seen that FD estimated by this method saturates at about 2.5. However, this method is faster than Pentland's [2] method since no Fourier transform computation is included.

Keller *et al.* [12] proposed a modification of a method due to Voss [14]. Let $P(m, L)$ be the probability that there are m intensity points within a box of size L centered about an arbitary point of image intensity surface. For any value of L we have

$$\sum_{i=1}^{N} P(m, L) = 1$$

where N is the number of possible points in the box of size L. In fact, N equals the number of spicels the box contains. Suppose that the image is of size $M \times M$. If one overlays the image intensity surface with boxes of side L, then the number of boxes needed to cover the whole image is

$$N_L = M^2 \sum_{i=1}^{N} (1/m)P(m, L).$$

Since M^2 is constant for an image, let it be dropped from the expression, i.e.,

$$N_L = \sum_{i=1}^{N} (1/m)P(m, L). \tag{5}$$

Using (4) and (5), we can estimate D. This method has the same limitation as in Gangepain and Roques-Carmes method. To avoid this, Keller *et al.* [8] devised a new version of probability estimation. In this refinement, the fractal surface between the center point of a box and its neighbors are approximated by linear interpolation. The newly interpolated surface is intersected with the box and number of points m in the box of side L is recorded. N_L is calculated using (5). This method takes a little more time, but gives satisfactory results except for the image intensity surfaces whose FD's are very high. Fig. 3 shows that FD estimated by this method saturates at 2.75.

III. Differential Box-Counting Approach and Experimental Results

We have a basic equation of FD given by

$$D = \frac{\log (N_r)}{\log (1/r)}.$$

In our proposed method, N_r is counted in a different manner from those in [11], [12]. Consider that the image of size $M \times M$ pixels has been scaled down to a size $s \times s$ where $M/2 \geq s > 1$ and s is an integer. Then we have an estimate of $r = s/M$. Now, as in previous techniques, consider the image as a 3-D space with (x, y) denoting 2-D position and the third coordinate (z) denoting gray level. The (x, y) space is partitioned into grids of size $s \times s$. On each grid there is a column of boxes of size $s \times s \times s'$. If the total number of gray levels is G then $\lfloor G/s' \rfloor = \lfloor M/s \rfloor$. For example, see Fig. 1, where $s = s' = 3$. Assign numbers 1, 2, $\cdots$ to the boxes as shown. Let the minimum and maximum gray level of the image in the (i, j)th grid fall in box number k and l, respectively. In our approach

$$n_r(i, j) = l - k + 1$$

is the contribution of N_r in (i, j)th grid. For example, in Fig. 1, $n_r(i, j) = 3 - 1 + 1$. (Although in this figure, for simplicity, smooth image surface is taken, but in reality it will be digital image surface.) Because of the differential nature of computing n_r we call our method the *differential box-counting* (DBC) approach. Taking contributions from all grids, we have

$$N_r = \sum_{i,j} n_r(i, j). \tag{6}$$

N_r is counted for different values of r, i.e., different values of s. Then using (3), we can estimate D, the fractal dimension, from the least square linear fit of log (N_r) against log $(1/r)$.

The reason for counting N_r in this manner is that it gives a better approximation to the boxes intersecting the image intensity surface, which is quantized in space and gray value. This is particularly so when there is sharp gray level variation in neighboring pixels in the image. In a sense, the DBC approach makes a digital approximation of thickness of the blanket that covers the image intensity surface at a particular resolution. Box counting in the other methods [11], [12] does not cover the image surface so well and hence, cannot capture the fractal dimension for rough textured surface. The other problems of box counting are discussed in Section IV.

A typical plot of log (N_r) versus log $(1/r)$ of the image D33 [15] is shown in Fig. 2. Let $y = mx + c$ be the fitted straight line, where y denotes log (N_r) and x denotes log $(1/r)$. Then error of fit E can be expressed as the root mean-square distance of the points from the fitted line.

$$E = \frac{\sqrt{\sum_{i=1}^{n} \frac{(mx_i + c - y_i)^2}{(1 + m^2)}}}{n}. \tag{7}$$

The error provides a measure of fit so that the lower the value of E, the better is the fit.

For our experiment we took 12 images from Brodatz [15] and 36 synthetic textured images. The synthetic images are actually noise added to an absolutely smooth image surface at gray level 128. Zero-mean Gaussian noise with different standard deviation σ has been added to this smooth image surface so that the resulting gray levels lie in the range 0–255. For all of these synthetic images, the size is equal to 128 × 128.

We choose five algorithms, including ours, for comparative study. The other four algorithms are due to Pentland [2], Peleg *et al.* [10], Gangepain and Roques-Carmes [11] and Keller *et al.* [12]. At first, the algorithms are tested on the synthetic images. It is expected that the FD will increase if the noise σ increases, and beginning at 2.0, the FD will asymptotically go towards a value of 3.0. A good method should reflect this desirable feature.

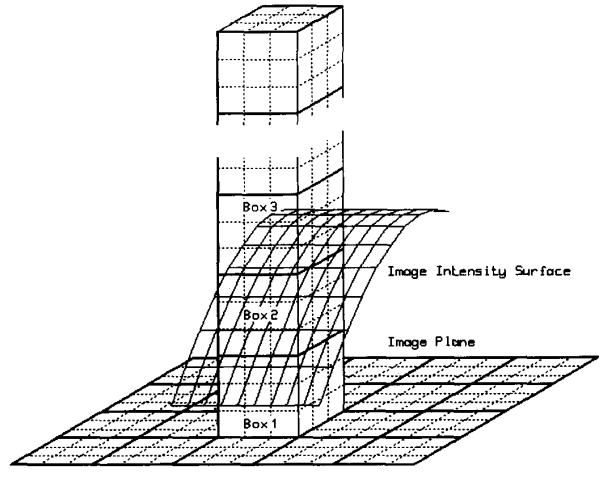


Fig. 1. Determination of n_r by proposed method.

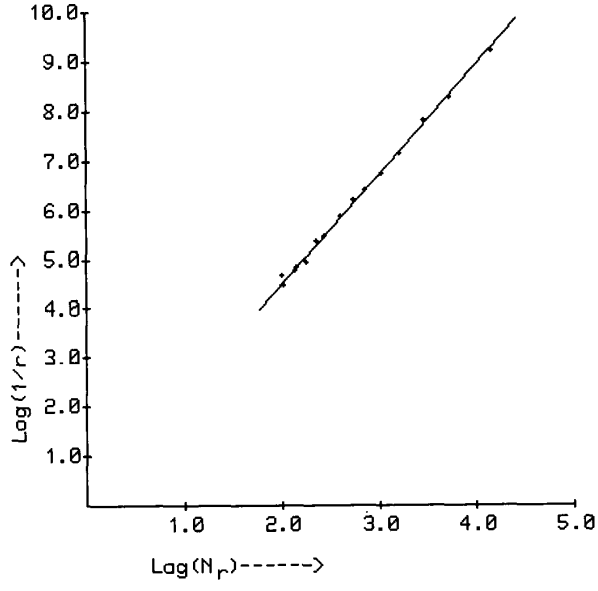
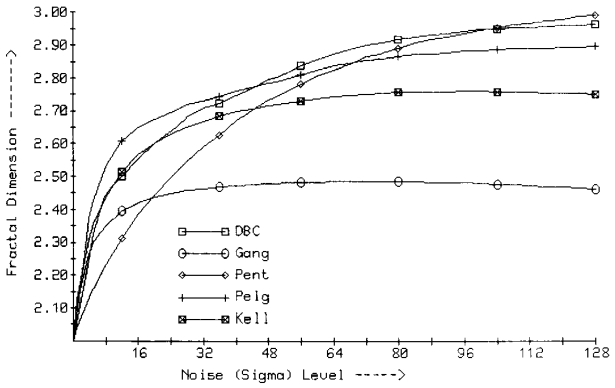


Fig. 2. Plot of log N_r versus log $(1/r)$ of texture image (D33 in Brodatz [15]).

The results are plotted in Fig. 3. It is seen that the methods due to Pentland, Peleg *et al.*, and the DBC approach give satisfactory results. On the other hand, the methods from Gangepain and Roques-Carmes' and Keller *et al.* give satisfactory result up to a certain level of roughness of the image intensity surface. After a certain value of σ, the slope of the curve nearly goes to zero, and hence, these methods do not cover the full dynamic range of FD. In the method due to Keller *et al*, the acceptable range of FD is 2.0–2.75, while in Gangepain and Roques-Carmes' method, the range is 2.0–2.5

Next we compare the computational complexity of different methods. Our method is readily comparable with methods from Gangepain and Roques-Carmes' [11] as well as Keller *et al.* [12] because in all these cases, the computations are done taking boxes



of different sizes. Relation between r in our method and L in methods from Gangepain and Roques-Carmes', as well as from Keller *et al.*, is given by $L = s = r \times M$, where $M \times M$ is the size of the image. The number of computations required in terms of M and r for each method is shown in Table I. The computational superiority of our method is readily observed from this table.

In the approach of estimates of finding FD using Peleg's [10] method, the upper and lower blanket are computed in each iteration. To calculate each blanket, one needs $10M^2$ comparisons, $6M^2$ additions, $6M^2$ subtractions, one multiplication and one division. Initially, one needs $4M^2$ subtractions to calculate the zeroth blanket, before the start of an iteration.

The computational complexity of Pentland's [2] method is very high. It needs Fourier transformation to find the Fourier power spectra. To calculate FFT of an $M \times M$ image by conventional algorithm one needs $2M \times \log_2 M$ operations, where each of such operations consists of $M + M \log_2 M$ additions, $M + M \log_2 M$ subtractions, $4M$ multiplications, three divisions, one sine and one cosine operation. As other overheads, one need $4M^2$ comparisons and M^2 square roots.

The computation required for regression is not considered in any of the methods because regression is common to all. A comparative study on actual number of operations required for real texture image (D03 of Brodatz [15]) is shown in Table II, where 15 iterations are taken in methods other than Pentland's. Note that iterations are not necessary in Pentland's method. Again, one can see that our method is computationally cheaper than other methods.

Next, we have taken a set of 12 texture images from Brodatz [15] album to compare the FD obtained by different methods. The images are shown in Fig. 4, while results are presented in Table III. It can be seen that methods from Pentland [2], from Peleg *et al.* [10] and from us give identical results. Also, it may be noted that the FD from our method is intermediate between that from Pentland and Peleg *et al.* for all texture images. Fig. 3 also shows that FD from our method is intermediate between that from Pentland and Peleg *et al.* in the range 2.23–2.58, which is roughly the range of FD for these texture images. Finally, the comparative study of error of fit shows that error in our method is less than that from Gangepain and Roques-Carmes [11], as well as from Keller *et al.* [12], but somewhat more than that from Pentland and Peleg *et al.* However, since the results are consistent throughout, the error has a negligible effect on the computed FD, which has been checked on 20 other texture images.

Computational complexity with respect to CPU time is shown graphically in Fig. 5 where the algorithms programmed in C language were run on a SUN 3/280 machine in a UNIX environment. For all other methods, due to Pentland [2], 15 iterations are considered. Pentland's method does not need any iteration. The method due to Gangepain and Roques-Carmes [11] needs about the same computation time as DBC, but is not suitable for estimating the FD since it does not cover even half the total dynamic range of FD. Among the others, it is seen that DBC is about 2.67 times faster than the method from Peleg *et al.* [10]. But according to [10], it needs 50 iterations to obtain a reasonable estimation of FD. If 50 iterations are considered instead of 15, then the method from Keller *et al.* [12] is faster than that from Peleg *et al.* Our method is about 5.33 times faster than the method in [12].

TABLE I
COMPARISON OF COMPLEXITY OF THREE METHODS

	DBC	Gang	Kell
Comparison	$2M^2$	$1/r^4$	r^2M^4
Addition	$4/r^2$	$M/r + 4/r^2$	$2M^2 + M^2L^2$
Subtraction	$2/r^2$		M^2
Multiplication	$3/r^2$	$2/r^2$	
Division	$3/r^2$	M^2	$r^3M^3 + M^2L^2$

TABLE II
COMPARISON OF NUMBER OF COMPUTATIONS

	DBC	Gang	Kell	Pelg	Pent
Comparison	464142	417040	2687181	2457600	45056
Addition	37524	445183	418935	1474560	933376
Subtraction	18762	0	144060	1540096	636928
Multiplication	28143	18762	0	15	825856
Division	28143	232071	261630	15	101888
Square Root	0	0	0	0	16384
Sine/Cos	0	0	0	0	3584

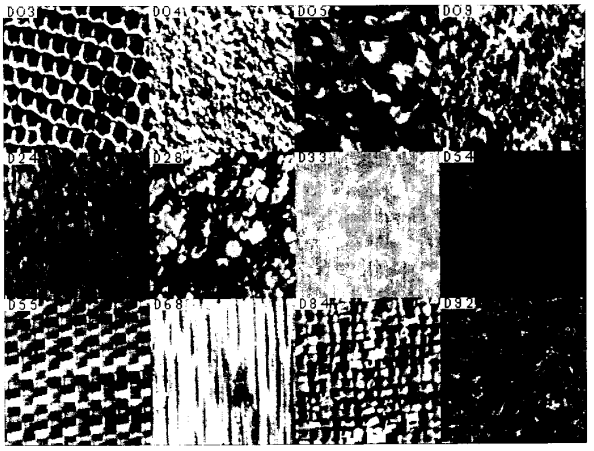


Fig. 4. Natural textures (Brodatz [15]).

IV. MODIFICATION AND DISCUSSION

Although the DBC method gives a very good estimate of FD, a possibility of error exists due to the quantization nature of the approach, especially when the image intensity surface is smooth. As an example, consider an image (S01 in Fig. 6) whose gray level

TABLE III
FRACTAL DIMENSION OF NATURAL TEXTURES USING DIFFERENT METHODS (IMAGE NUMBERS CORRESPOND TO BRODATZ'S BOOK)

Texture Images	DBC		Pent		Pelg		Gang		Kell	
	FD	E	FD	E	FD	E	FD	E	FD	E
D03	2.60	0.032	2.54	0.004	2.69	0.011	2.40	0.074	2.63	0.036
D04	2.66	0.026	2.55	0.012	2.72	0.008	2.45	0.072	2.68	0.030
D05	2.45	0.032	2.38	0.003	2.52	0.012	2.38	0.057	2.57	0.025
D09	2.59	0.028	2.49	0.003	2.65	0.009	2.43	0.066	2.65	0.026
D24	2.45	0.022	2.36	0.002	2.59	0.007	2.39	0.048	2.57	0.014
D28	2.55	0.033	2.48	0.007	2.61	0.012	2.41	0.066	2.62	0.031
D33	2.23	0.007	2.21	0.002	2.34	0.003	2.26	0.024	2.36	0.008
D54	2.39	0.023	2.31	0.011	2.53	0.008	2.35	0.044	2.51	0.015
D55	2.48	0.031	2.37	0.006	2.60	0.010	2.39	0.057	2.59	0.023
D68	2.52	0.024	2.44	0.008	2.63	0.007	2.40	0.054	2.60	0.019
D84	2.60	0.029	2.47	0.001	2.68	0.009	2.43	0.067	2.65	0.028
D92	2.50	0.023	2.38	0.007	2.59	0.007	2.41	0.052	2.59	0.018

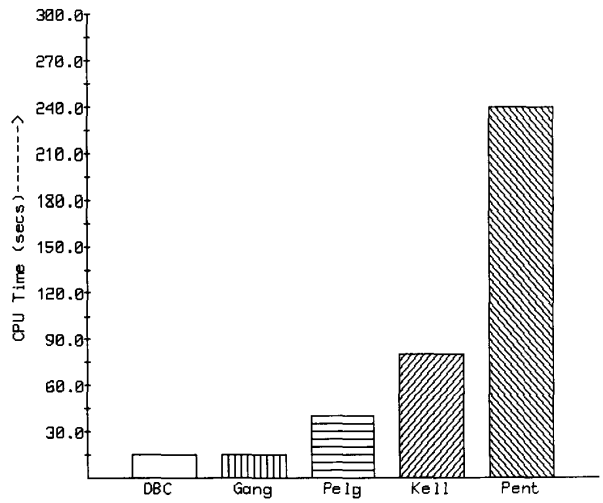


Fig. 5. CPU time required in different techniques. (Methods in [11] (Gang) and [12] (Kell) cover about 50 and 75% of the total dynamic range of FD.)

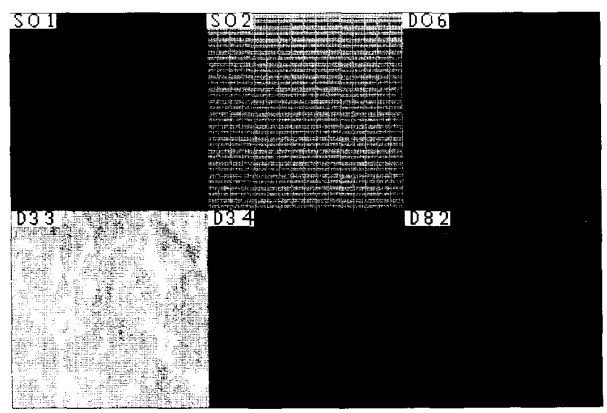


Fig. 6. Smooth textures.

range is between 116 and 119. If we directly use the DBC method we note that as long as $28 \geq s > 1$, all the gray values in this range will fall in a single box and estimated FD will be 2.0, which

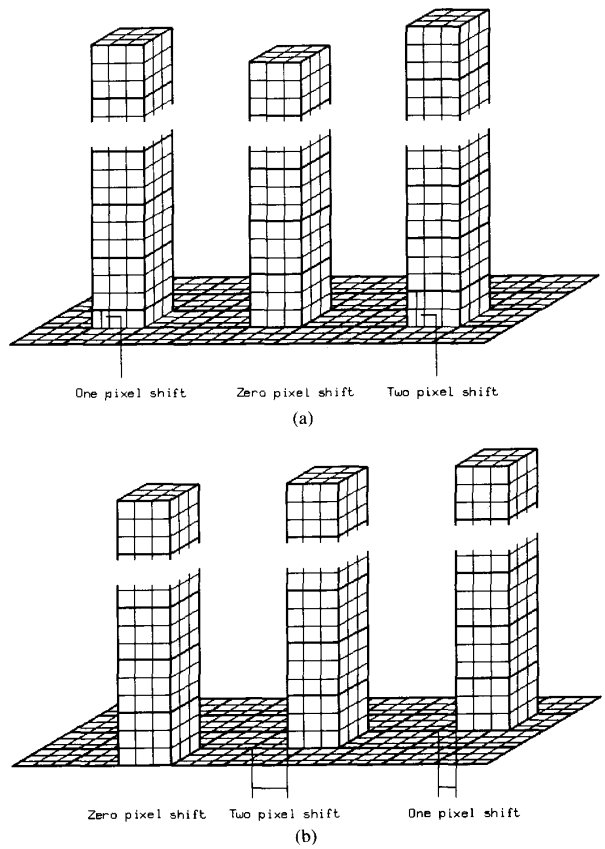


Fig. 7. (a) Vertical shift. (b) Horizontal shift.

in principle should be (slightly) more than 2.0. The same error will be introduced by other box-counting approaches as well.

We propose a modification that can take care of this problem. In the original DBC approach, the column of boxes are placed so that the height of jth box in ith column is the same as the height of jth box in any other column. In the modified approach, the columns are given random shift in the vertical, i.e., z-direction so that the height of the jth box in two columns may not be the same. The random shifts are given in the multiples of gray level height. See, for example, Fig. 7(a). It may be understood that the small varia-

TABLE IV
COMPARISON OF COMPLEXITY WITH OR WITHOUT SHIFT

	Computational Complexity of DBC's		
	Without Shift	Predetermined Shift	Random Shift
Comparison	$2M^2$	$2M^2$	$2M^2$
Addition	$4/r^2$	$6/r^2$	$6/r^2$
Subtraction	$2/r^2$	$2/r^2$	$2/r^2$
Multiplication	$3/r^2$	$4/r^2$	$4/r^2$
Division	$3/r^2$	$4/r^2$	$4/r^2$
Rand-gen[a]			$1/r^2$

[a]Random number generation.

TABLE V
FRACTAL DIMENSION OF TEXTURES WITH OR WITHOUT SHIFT

Texture Images	FD Estimated by Using DBC		
	Without Shift	Predetermined Shift	Random Shift
D06	2.32	2.36	2.35
D33	2.26	2.31	2.30
D34	2.20	2.26	2.25
D82	2.46	2.48	2.48
S01	2.00	2.19	2.16
S02	2.17	2.32	2.31

Gang: Method from Gangepain and Roques-Carmes [11].
Kell: Method from Keller *et al.* [12].
Pent: Method from Pentland [2].
Pelg: Method from Peleg *et al.* [10].

tion of gray value in image can be captured by a modified approach. In fact, as seen from Table V, the image S01 whose FD was estimated as 2.0 now has a modified FD = 2.16 by the random shifted box placement.

Like random shift in z-direction, there is a need for placement of the column of boxes by a shift in plane, i.e., in x or y direction. See, for example, Fig. 7(b). Such a necessity may be explained by the synthetic image of S02 of Fig. 6. In this case, the spatial period of variation in gray level can beat with the spatial period of box columns so that any gray level difference does not fall in a single column of box, and hence is not counted. The beating effect can be overcome by an x and/or y shift of columns of boxes. The shifts are in the multiples of pixel length/breadth. In Table V, it is seen that FD without a shift is estimated to be 2.17 changes to 2.32 with shift.

The random shifting of columns needs some additional computation involving random number generation and differential box counting in each column. Instead of random shifts, we can make predetermined shifts that reduce the computational overhead to a great extent. The difference in FD estimation by predetermined and random shifts are seen to be small in Table V, whereas computational complexity is shown in Table IV.

Experiment has been done, not only on synthetic images but also on real textured images. Results on some smooth textured images from Brodatz [15] are shown in Fig. 6 and in Table V.

V. CONCLUSION

A simple, accurate, and efficient differential box-counting approach has been proposed to estimate fractal dimension of image. The approach can be readily extended to a three-dimensional image as well. The modification of the basic approach suggested in Section IV is useful in smoothly textured images. It is expected that such a modification will also give a better estimation for other box-counting approaches [11], [12].

ACKNOWLEDGMENT

The authors wish to thank Prof. D. Dutta Majumder for his interest in this work and Prof. A. K. Jain of Michigan State University for supplying texture images.

REFERENCES

[1] B. B. Mandelbrot, *Fractal Geometry of Nature*. San Francisco: Freeman, 1982.
[2] A. P. Pentland, "Fractal based description of natural scenes," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-6, pp. 661–674, 1984.
[3] A. P. Pentland, "Shading into texture," *Artificial Intell.*, vol. 29, pp. 147–170, 1986.
[4] J. P. Rigaut, "Automated image segmentation by mathematical morphology and fractal geometry," *J. Microscopy*, vol. 150, pp. 21–30, 1988.
[5] J. Orford and W. Whalley, "The use of fractal dimension to characterize irregular-shaped particle," *Sedimentology*, vol. 30, pp. 655–668, 1983.
[6] B. H. Kaye, "Fractal dimension and signature wave form characterization of fine particle shape," *Amer. Lab.*, pp. 55–63, 1986.
[7] M. Bernsley, *Fractal Everywhere*. San Diego, CA: Academic, 1988.
[8] A. Jacquin, "A fractal theory of iterated Markov operators, with applications to digital image coding," Ph.D. dissertation, Dept. Math., Georgia Inst. Technol., 1989.
[9] G. Zorpette, "Fractal: Not just another pretty picture," *IEEE Spectrum*, pp. 29–31, 1988.
[10] S. Peleg, J. Naor, R. Hartley, and D. Avnir, "Multiple resolution texture analysis and classification," *IEEE Trans. Pattern Anal. Macine Intell.*, vol. PAMI-6, pp. 518–523, 1984.
[11] J. Gangepain and C. Roques-Carmes, "Fractal approach to two dimensional and three dimensional surface roughness," *Wear*, vol. 109, pp. 119–126, 1986.
[12] J. Keller, R. Crownover, and S. Chen, "Texture description and segmentation through fractal geometry," *Comput. Vision Graphics and Image Processing*, vol. 45, pp. 150–160, 1989.
[13] R. Creutzberg and E. Ivanov, "Computing fractal dimension of image segments," in *Proc. III Int. Conf. Comput. Anal. of Images and Patterns,. CAIP'89*, 1989.
[14] R. Voss, "Random fractals: Characterization and measurement," in *Scaling Phenomena in Disordered Systems*, R. Pynn and A. Skjeltorp, Eds. New York: Plenum, 1986.
[15] P. Brodatz, *Texture: A Photographic Album for Artists and Designers*. New York: Dover, 1966.

Visual Field Information in Low-Altitude Visual Flight by Line-of-Sight Slaved Helmet-Mounted Displays

Arthur J. Grunwald and Silvia Kohn

***Abstract*—The pilot's ability to derive control-oriented visual field information from teleoperated helmet-mounted displays in nap-of-the-earth flight is investigated in this paper. The visual field with these**

Manuscript received May 4, 1991; revised September 10, 1992 and March 5, 1992. This work was supported by a grant from NASA Ames Research Center, Aerospace Human Factors Research Division, Moffett Field, CA 94035, under cooperation agreement No. NAGW-1128. An earlier version of this paper was presented at the SPIE/SPSE Symposium on Electronic Imaging: Science and Technology, San Jose, CA, February 1991.

The authors are with the Faculty of Aerospace Engineering, Technion—Israel Institute of Technology, Haifa 32000, Israel.

IEEE Log Number 9212944.