A New Extended Differential Box-Counting Method by Adopting Unequal Partitioning of Grid for Estimation of Fractal Dimension of Grayscale Images

Soumya Ranjan Nayak, Jibitesh Mishra and Rajalaxmi Padhy

Abstract Fractal dimension (FD) is most useful research topic in the field of fractal geometry to evaluate surface roughness of digital images by using the concept of self-similarity, and the FD value should lie between 2 and 3 for surfaces of digital images. In this regard, many researchers have contributed their efforts to estimate FD in the digital domain as reported in many kinds of the literature. The differential box-counting (DBC) method is a well-recognized and commonly used technique in this domain. However, based on the DBC approach, several modified versions of DBC have been presented like relative DBC (RDBC), improved box counting (IBC), improved DBC (IDBC). However, the accuracy of an algorithm for FD estimation is still a great challenge. This article presents an improved version of DBC algorithm by partitioning the box of grid into two asymmetric patterns for more precision box count and provides accurate estimation of FD with less fit error as well as less computational time as compared to existing method like DBC, relative DBC (RDBC), improved box counting (IBC), and improved DBC (IDBC).

Keywords Fractal dimension · DBC · RDBC · SDBC · IBC · IDBC

S. R. Nayak (✉) · R. Padhy
Department of Information Technology, College of Engineering and Technology, Bhubaneswar 751003, Odisha, India
e-mail: nayak.soumya17@gmail.com

R. Padhy
e-mail: padhyrajalaxmi123@gmail.com

J. Mishra
Department of Computer Science and Application, College of Engineering and Technology, Bhubaneswar 751003, Odisha, India
e-mail: jmishra@cet.edu.in

A. K. Nandi et al. (eds.), *Computational Signal Processing and Analysis*, Lecture Notes in Electrical Engineering 490,
https://doi.org/10.1007/978-981-10-8354-9_5

1 Introduction

Fractal geometry was popularized by Mandelbrot [1] to interpret and characterize complex shapes and surfaces with self-similarity in nature, like mountain, coastlines, and clouds. The concept self-similarity is most important to describing the textures images and useful for estimation of fractal dimension (FD) of complicated objects found in nature, which is failed to analyze by the traditional Euclidean geometry. Fractal dimension can be used as a most important feature in the field of image processing to characterize roughness, smoothness, and self-similarity of natural images. Pentland [2] popularized about smoothness versus roughness of image surface, where the smooth image indicates the fractal dimension of two and rough image as three. In this regard, various researchers have developed many box-counting algorithms. Fractal geometry has provided a numerical model for various complex and irritated objects that initiate in nature [1–3], in terms of coastline, mountains, and clouds. Fractal dimension (FD) is also useful in several fields of application that including analysis of texture and texture segmentation [4–6], shape measurement and texture classification [7], the study of the image as well as graphic analysis in other fields [8–10]. Various popular fractal dimension theories have been proposed and suggested by many researchers for grayscale and color images. Gangepain has presented the well-known reticular box-counting algorithm [11], and Keller et al. presented probabilistic theories [12].

Sarkar and Chaudhuri [4, 13] studied five different algorithms in box counting on synthetic images and suggested the proficient differential box-counting (DBC) algorithm to computing fractal dimension of images. Their results have shown that Keller et al. [12] give the satisfactory result, after reaching a particular level of noise (s), and image intensity surface with the slope of the curve tends to zero. As the natural scenes rather reveal some statistical self-similarity, not deterministic self-similarity, therefore both the methods do not estimate the dynamic range of FD; the reduction factor r was introduced so that if an object is broken down with a factor r in all n dimensions, then we can call as statistically alike to the original one to satisfy Eq. (1) that described in Sect. 2.

DBC algorithm was considered to be an efficient method presented in [14, 15]. However, in the case of DBC method there are three major drawbacks reported by Li et al. [16], i.e., selection of proper height of the box, calculation of exact box number and partition of image plane. In order to avoid this situation, the same author presented another method called an improved box-counting method [16] in order to avoid the three above-mentioned drawbacks of DBC method. Again some demerits found in DBC method are reported by Liu et al. [17] that over-counting and under-counting problem found while executing the method. To avoid such situation, the same author [17] presented another method called improved differential box-counting method. Woraratpanya [18] proposed triangle box counting (TBC) to improve the exactness, though this algorithm is only meant for binary images. Recently, an improved triangle box-counting method (ITBC) was proposed by Kaewaramsri and Woraratpanya [19] based on DBC approach to solve

over-counting problem by implementing triangle box partition. Nayak et al. [20] presented simple and proficient improved version of DBC algorithm for more accurate FD estimation of grayscale images. Another modified DBC method also presented by same author Nayak et al. [21] to avoid shortcomings like over-counting and under-counting and simultaneously provides less fitting error for grayscale images.

2 Basic Principle of Fractal Dimension and Related Work

There are several dimensions to express fractal dimensions in the field of fractal geometry, such as Hausdorff dimension, information dimension, packing dimension, box-counting dimension. Among these dimensions, box-counting dimension is most popular and widely used techniques in fractal geometry to express the concept of self-similarity. Fractal dimension can be evaluated by using equation below:

$$D = \log(N)/\log(1/r) \tag{1}$$

2.1 DBC Approach

Sarkar and Chaudhuri [13] projected the differential box-counting (DBC) method for estimating FD of grayscale images. In order to implement this method, they represent grayscale image in $3D$ space, where $2D$ space like (x, y) represents an image plane and third coordinates like z represent the gray level. For this experimental analysis, they took a square image of size $M \times M$ and partitioned into $L \times L$ grids. Each and every grid comprises a stake of boxes of size $L \times L \times H$, where H indicates the height of every box, and this height can be calculated in terms of $L \times G/M$, where G represents the total number of gray levels. Let the maximum and minimum gray values of (i,j)th grid fall in Lth and Kth box, respectively, and then the box count $n_r(i,j)$ can be calculated as follows:

$$n_r(i,j) = L - K + 1 \tag{2}$$

By taking involvement from all blocks, N_r is counting for different values of L as follows:

$$N_r = \sum_{i,j} n_r(i,j) \tag{3}$$

2.2 RDBC Approach

Based on original DBC, Jin and Jayasooriah [22] presented an improved version of DBC called relative DBC (RDBC) by adopting same maximum and minimum intensity point on the grid and taking the scale limit such as higher and smaller limits of scale ranges for accurate evaluation of FD of texture images. Finally, N_r can be evaluated as follows:

$$N_r = \sum_{i,j} \text{ceil}[k \times ((K - L)/L')] \tag{4}$$

where k represents the coefficient in z-direction and ceil(.) is used to set the nearest integer.

2.3 IBC Approach

In similar like DBC and RDBC, Li et al. [16] presented another improved DBC mechanism by adopting three major parameters like the selection of box height, box number estimation, and partition of intensity surface. They are selecting box height by using the formula as follows:

$$r' = \frac{L}{1 + 2a\sigma} \tag{5}$$

where a is a positive integer and set the appropriate value as 3, σ represents standard deviation, and $2a\sigma$ represents image roughness. Finally, $n_r(i,j)$ are evaluated as follows

$$n_r(i,j) = \begin{cases} \text{ceil}\left(\frac{K-L}{r'}\right) & \text{if } K \neq L \\ 1 & \text{otherwise} \end{cases} \tag{6}$$

N_r can be calculated by taking the contribution of all grids.

2.4 IDBC Approach

Liu et al. [17] proposed improved version of DBC algorithm called improved differential box-counting method (IDBC) for estimating FD of grayscale image. In their proposed method, three modifications have been done such as modifying box counting, recalibrating box of block in spatial plane of (x, y) direction, and choosing

proper grid box size and final $n_r(i,j)$ was calculated by taking the maximum contribution from original grid and shifted grid by using Eq. (2)

$$n_r(i,j) = \begin{cases} \text{ceil}\left(\frac{I_{\max} - I_{\min} + 1}{s'}\right) I_{\max} I_{\min} \\ 1 \quad \text{otherwise} \end{cases} \tag{7}$$

N_r can be calculated by taking the contribution of all grids, and finally, the FD can be estimated for all these methods from the least square regression line of $\log(N_r)$ verses $\log(1/r)$.

After analyzing above-mentioned methods, we conclude that the improvement technique is implemented to evaluate FD based on two major issues, namely over-counting and under-counting problems. Most of the techniques [22, 16] attempt to solve only under-counting by implementing box height. On the other hand, the technique [19] attempts to solve over-counting problems and few techniques like [17, 21] tried to solve both the problems. All these methods provide more fitting error as well as more computational time; hence, no proper box-counting technique is presented. Therefore, this article presented more accurate estimation by partitioning the box of the grid into two asymmetric patterns for more precision box count and yields more accurate estimation in terms of less fit error as well as less computational time.

3 Proposed Methodology

In this section, our improved proposed methods are discussed. The improvement can be achieved by partitioning the grid box into two asymmetric patterns for more precision box count, which is represented in Fig. 1.

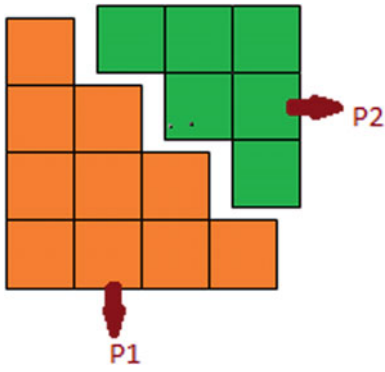

Fig. 1 Sktech of partitioning grid into two asymmetric pattern

3.1 Algorithm Details

In this section, the detailed algorithm of our proposed method is summarized below.

1. Read the image of size $M \times M$.
2. Divide the image into the box of size $L \times L$, where L varies from 2 to $M/2$.
3. Estimate height of grid $H = L \times G/M$, and each box size should be $L \times L \times H$, where G represents total gray level.
4. Partitioning the box of grid into two asymmetric patterns such as $p1$ and $p2$ based on step 3.
5. Determine $n_r(i,j)$ for each pattern in Fig. 1 based on Eq. (8), where max and min represents maximum and minimum intensity point at each pattern and NP represents total no of pixel points present at each pattern.

$$\begin{aligned} P_1 &= \text{ceil}(((\max - \min) + 1)/H) \times (\text{NP}/(L \times L)) \\ P_2 &= \text{ceil}(((\max - \min) + 1)/H) \times (\text{NP}/(L \times L)) \end{aligned} \tag{8}$$

6. Final $n_r(i,j)$ can be calculated by taking maximum contribution from each pattern using Eq. (9).

$$n_r(i,j) = \max\{P_1, P_2\} \tag{9}$$

7. For the distinct value of L, the entire amount of boxes required to wrap entire image surface is evaluated using Eq. (3).
8. Plot the least square linear fit of $\log(N_r)$ versus $\log(1/r)$ to estimate FD from negative slope of the fitted straight line.

4 Experimental Setup

In order to evaluate and compare the performance of our proposed method, we are considering four well-known methods such as DBC, RDBC, IBC, and IDBC. All these algorithms are implemented in MATLAB (R2014a, 64 bit) with the standard environment (Windows 8, 64-bit operating system, Intel (R) i7–4770 CPU @ 3.40 GHz). The experimental setup carried out on three experiments, which have a set of original 16 real brodatz images [23] presented in Fig. 2, one set of twelve generated smooth texture image represented in Fig. 6, one set of fifty generated synthetic images by different gray level shifts, the precision of FD estimation algorithm are express by means of fitting error as follows in Eq. (10)

$$\text{Error fit} = \frac{1}{n}\sqrt{\sum_{i=1}^{n} \frac{\mathrm{d}x_i + c - y_i}{1 + d^2}} \tag{10}$$

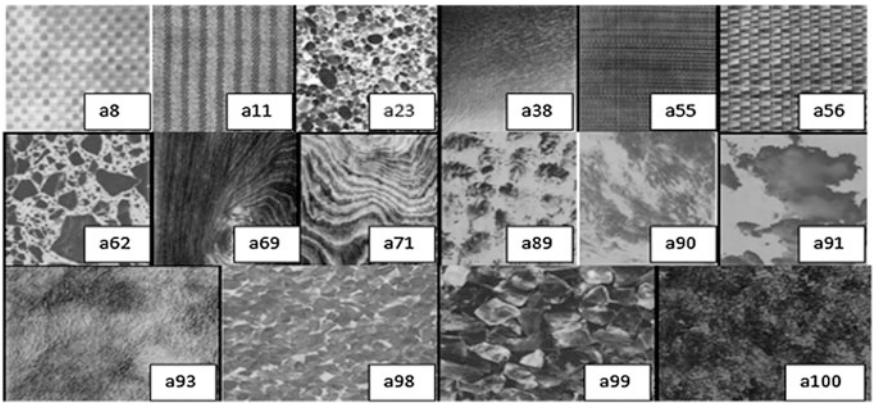

Fig. 2 Sixteen Brodatz database texture images

4.1 Test on Real Texture Images

In this section, we are using a set of 16 real texture images [23] from the Brodatz database of size 256 × 256 for our experimental analysis which is presented in Fig. 2. In this experimental setup, the FD generated from DBC algorithm are in the range of 2.19 to 2.67 similarly, the other measure like RDBC, IBC, IDBC and proposed method are range from 2.28 to 2.73 and 2.29 to 2.73 and 2.30 to 2.72 and 2.34 to 2.77 which are represented in Table 1 and error fit are listed in Table 2 and graphical representation are presented in Figs. 3 and 4 respectively. The average error fit is estimated from each method like DBC, RDBC, IBC, IDBC, and proposed method are 0.0418, 0.0669, 0.0681, 0.0453, 0.0419 respectively are listed in Table 5 and presented in Fig. 5. From this experimental result, it is crystal clear that the fitting error from proposed method yields less average error fit as compared to other existing four chosen methods as well as less fitting error for individual image presented in Fig. 4, and provides less computational time presented in Table 4 and Fig. 6.

4.2 Test on Synthetic Images1

In this section, we have generated fifty transformed images based on Eq. (11). Resultant generated images are achieved from original a100 Brodatz images.

$$\mathrm{RTI}(i,j) = \mathrm{round}\left(\frac{I(i,j) - \min(i,j)}{\max(i,j) - \min(i,j)} \times W\right) + (G/2 - 2) \tag{11}$$

where RTI is the resultant transformed image generated from original image $I(i,j)$, $round(.)$ is used for rounding to nearest integer, $\max(i,j)$ and $\min(i,j)$ are

Table 1 Computational FD of Brodatz images

Image name	Fractal dimension				
	DBC	RDBC	IBC	IDBC	Proposed
a8	2.27	2.36	2.39	2.40	2.43
a11	2.59	2.67	2.68	2.66	2.71
a23	2.58	2.61	2.62	2.63	2.67
a38	2.51	2.58	2.60	2.57	2.61
a55	2.67	2.73	2.73	2.72	2.77
a56	2.53	2.60	2.61	2.63	2.65
a62	2.50	2.55	2.57	2.55	2.58
a69	2.51	2.53	2.55	2.56	2.60
a71	2.54	2.54	2.56	2.58	2.62
a89	2.42	2.49	2.52	2.50	2.54
a90	2.30	2.43	2.45	2.47	2.51
a91	2.19	2.28	2.29	2.30	2.34
a93	2.60	2.65	2.67	2.66	2.70
a98	2.42	2.50	2.51	2.46	2.50
a99	2.41	2.48	2.49	2.48	2.52
a100	2.57	2.66	2.67	2.61	2.66

Table 2 Computational EF of Brodatz images

Image name	Error fit				
	DBC	RDBC	IBC	IDBC	Proposed
a8	0.0686	0.0757	0.0779	0.0665	0.0623
a11	0.0525	0.0550	0.0561	0.0449	0.0412
a23	0.0655	0.0677	0.0681	0.0585	0.0559
a38	0.0453	0.0478	0.0496	0.0356	0.0332
a55	0.0497	0.0517	0.0521	0.0400	0.0376
a56	0.0658	0.0683	0.0688	0.0581	0.0573
a62	0.0664	0.0678	0.0695	0.0559	0.0528
a69	0.0558	0.0545	0.0556	0.0459	0.0422
a71	0.0633	0.0617	0.0630	0.0554	0.0519
a89	0.0648	0.0671	0.0687	0.0563	0.0524
a90	0.0570	0.0675	0.0695	0.0630	0.0584
a91	0.0665	0.0756	0.0751	0.0644	0.0590
a93	0.0492	0.0474	0.0498	0.0383	0.0335
a98	0.0647	0.0691	0.0698	0.0546	0.0499
a99	0.0665	0.0690	0.0699	0.0589	0.0552
a100	0.0534	0.0569	0.0576	0.0422	0.0387

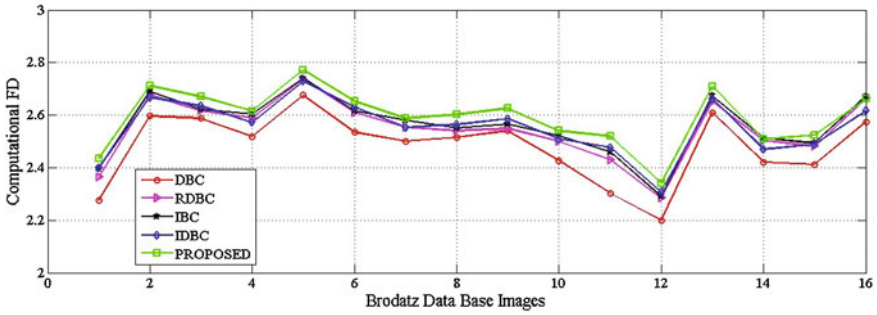

Fig. 3 Computational FD of sixteen Brodatz database images

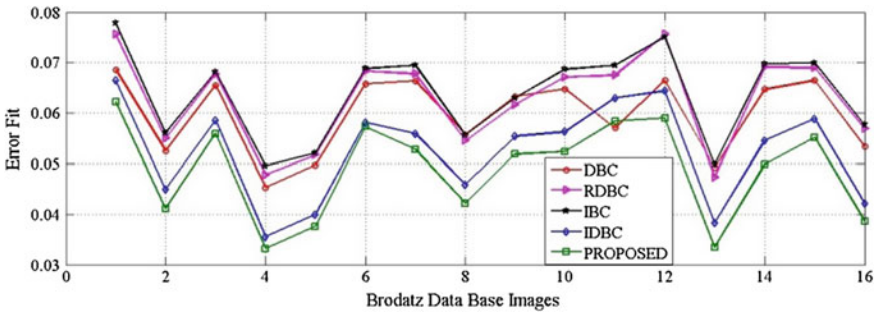

Fig. 4 Computational error fit of sixteen Brodatz database images

Table 3 Computational FD of twelve generated synthetic images

Image name	Fractal dimension				
	DBC	RDBC	IBC	IDBC	Proposed
s1	2.42	2.49	2.52	2.50	2.54
s2	2.42	2.49	2.52	2.50	2.54
s3	2.42	2.49	2.52	2.50	2.54
s4	2.42	2.49	2.52	2.50	2.54
s5	2.42	2.49	2.52	2.50	2.54
s6	2.42	2.49	2.52	2.50	2.54
s7	2.43	2.49	2.52	2.50	2.54
s8	2.43	2.49	2.52	2.50	2.54
s9	2.43	2.49	2.52	2.50	2.54
s10	2.43	2.49	2.52	2.50	2.54
s11	2.44	2.49	2.52	2.50	2.54
s12	2.44	2.49	2.52	2.50	2.54

Table 4 Computational time estimation

Images	Computational time				
	DBC	RDBC	IBC	IDBC	Proposed
Brodatz	69.803	42.184	398.499	112.897	53.051
Synthetic1	24.246	18.834	129.339	38.002	17.682
Synthetic2	16.817	10.204	102.438	26.156	12.655

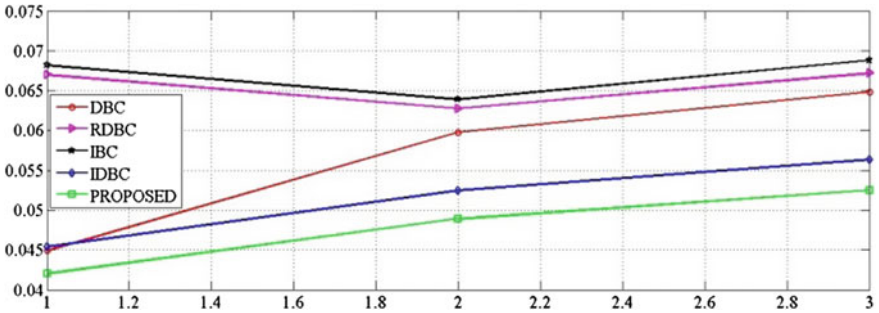

Fig. 5 Computational average error fit

represented maximum and minimum intensity point of original image, W is weighted coefficient varies from $1 \leq W \leq 50$, and G represents gray level; we estimate FD based on different W values, as the value of W increases simultaneously FD also increases and vice versa. Figure 7 represents the estimated FD of fifty generated synthetic images, from this experimental analysis.

We have seen that our proposed methodology yields better FD estimation than other competent methods. As we mention earlier that when the value of W increased means the generated FD also increased accordingly, while method like DBC, RDBC and IBC are failed to estimate accurate FD. When W value ranges from 1 to 21, the FD value from DBC yields FD less than 2 and, and similarly in case of RDBC the FD value also suddenly decreased *W27* to *W28* images. But in case of IBC the FD value was evaluated as greater than three in the first image that is *W1* image and gradually decrease when the W value is increased, while our proposed method provides consistent result in terms of increasing order as increased of W presented on Fig. 5, from above point of view it is crystal clear that our proposed method provides the wider range of FD.

4.3 Test on Synthetic Images2

In this experimental setup, a set of 12 synthetic images are generated, which are presented in Fig. 8. In this process, each image is generated from original image by incrementing each intensity point (by formula $2 \times (k - 2)$, where k lies from 1 to

12 in such a way that at the *k*th of 12th image, the maximum gray level should not exceed 255, and therefore alternatively we can say that the entire pixel surface is recalibrated to particular position of gray level against the original one in *z*-direction. Theoretically, it was clear that if we either step up or step down the intensity value with a constant number, then the FD should stay same as original one because theoretically both have the equal degree of roughness. From this experimental result presented in Table 3, we have seen that except DBC all other methods provide same estimated value. In DBC, the gray level is established on the particular area in the *z*-direction; the first box is in zero gray level position, and the utmost box is in 255 gray level position; this may cause over-counting but in case of proposed method. In this experimental analysis, our proposed method provides accurate estimation with less fit error and less computational time listed in Tables 4 and 5 respectively.

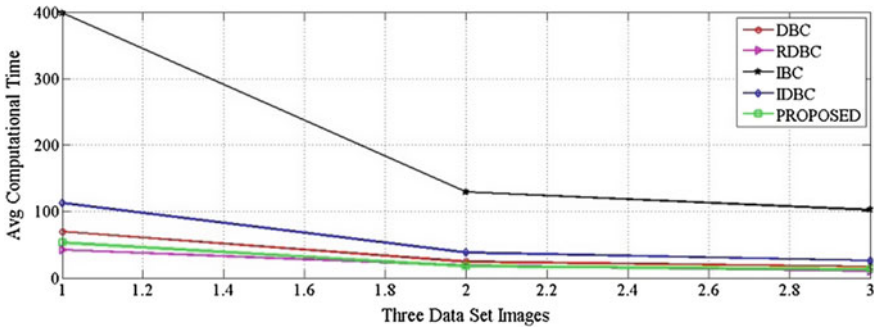

Fig. 6 Estimation of average computational time

Table 5 Average error fit estimation

Images	Average error fit				
	DBC	RDBC	IBC	IDBC	Proposed
Brodatz	0.0448	0.0669	0.0681	0.0453	0.0419
Synthetic1	0.0597	0.0627	0.0638	0.0524	0.0488
Synthetic2	0.0648	0.0671	0.0687	0.0563	0.0524

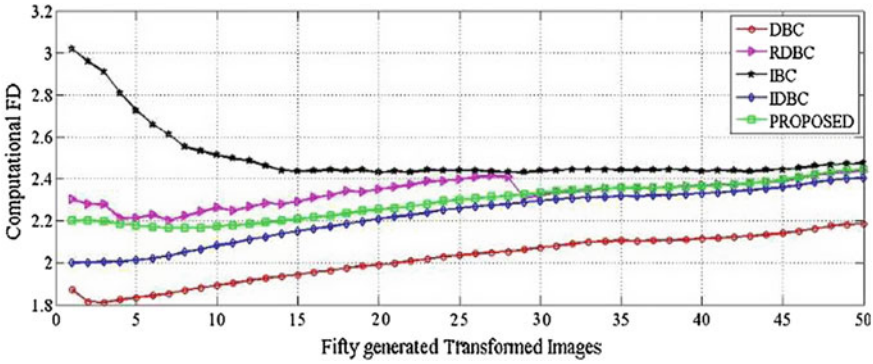

Fig. 7 Fractal dimension estimation of fifty generated synthetic images

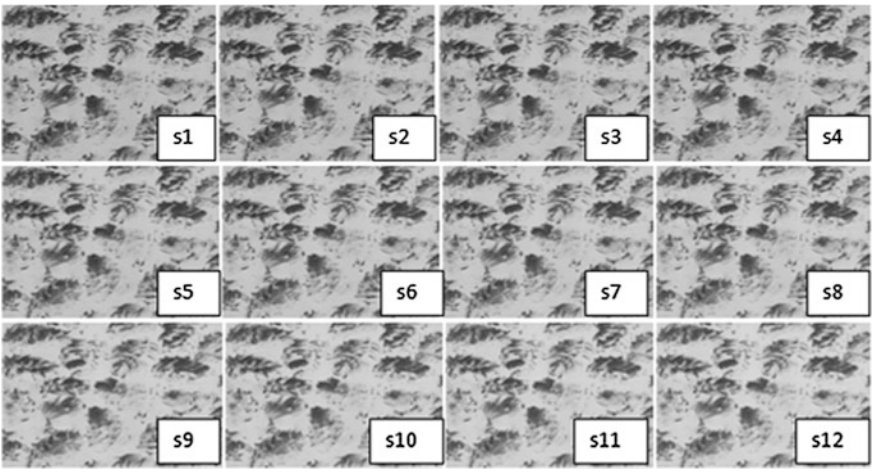

Fig. 8 Twelve generated synthetic images

5 Conclusion

In this article, we presented improved differential box-counting by partitioning the box of the grid into two asymmetric patterns for accurate estimation of FD of grayscale images. Three experiments were carried out to test our proposed method, and the experiment results are compared with our well-known methods such as DBC, RDBC, IBC, and IDBC. The result is generated from Brodatz database, and synthetic images show that the proposed method eradicated with maximum efficiency in terms of less fitting error for each image and also produced average less fitting error as compared to other chosen methods. It shows that our modified

method covers all objects with the wider range of fractal dimension as compared to existing methods. It is a robust and more precise method.

Acknowledgements The authors are sincerely thankful to the Department of Information Technology, College of Engineering and Technology, Bhubaneswar. And we are also thankful to all the authors of references.

References

1. Mandelbrot BB (1982) The fractal geometry of nature. Freeman, San Francisco
2. Pentland AP (1984) IEEE Trans Pattern Anal Mach Intell 6
3. Peitgen HO, Jurgens H, Saupe D (1992) Chaos and fractals: new frontiers of science, 1st edn. Springer, Berlin
4. Chaudhuri BB, Sarkar N (1995) IEEE Trans Pattern Anal Mach Intell 17
5. Liu S, Chang S (1997) IEEE Trans Image Process 6 (1997)
6. Ida T, Sambonsugi Y (1998) IEEE Trans Circ Syst Video Technol 8
7. Neil G, Curtis KM (1997) Pattern Recogn 30
8. Lin KH, Lam KM, Siu WC (2001) IEEE Proceedings on vision, image and signal processing 148
9. Asvestas P, Matsopoulos GK, Nikita KS (1998) J Vis Commun Image Represent 9
10. Bisoi AK, Mishra J (2001) Pattern Recogn Lett 22
11. Gangepain J, Carmes CR (1986) Wear 109
12. Keller J, Crownover R, Chen S (1989) Comput Vis Graph Image Process 45
13. Sarkar N, Chaudhuri BB (1994) IEEE Trans Syst Man Cybern 24
14. Wenlu X, Weixin X (1997) Chin Signal Process J 13
15. Yu L, Zhang D, Wang K, Yang W (2005) Pattern Recogn 38
16. Li J, Du Q, Sun C (2009) Pattern Recogn 42
17. Liu Y, Chen L, Wang H, Jiang L, Zhang Y, Zhao J, Wang D, Zhao Y, Song Y (2014) J Vis Commun Image Represent 25
18. Woraratpanya K, Kakanopas D, Varakulsiripunth R (2012) ASEAN Eng J Part D 1
19. Kaewaramsri Y, Woraratpanya K (2015) Recent advances in information and communication technology. In: Proceedings of the 11th international conference on computing and information technology, July 2–3, Bangkok, Thailand
20. Nayak SR, Mishra J, Padhy R (2016) Signal processing, communication, power and embedded system. In: Proceeding of the IEEE international conference, Oct 3–5, Paralakhemundi, India
21. Nayak SR, Mishra J, Padhy R (2016) Int J Comput Sci Inf Secur 14
22. Jin XC, Jayasooriah SH (1995) Pattern Recogn Lett 16
23. Brodatz P (1966) Texture: a photographic album for artists and designers, New York