Int. j. inf. tecnol. (March 2018) 10(1):39–47
https://doi.org/10.1007/s41870-017-0062-3

ORIGINAL RESEARCH

Fractal analysis of image sets using differential box counting techniques

Soumya Ranjan Nayak[1] · Jibitesh Mishra[2] · Pyari mohan Jena[3]

Received: 9 May 2017 / Accepted: 14 November 2017 / Published online: 30 November 2017

Abstract The fractal dimension (FD) is most useful technique to evaluate surface roughness of digital images in terms of gray scale and color images. A number of techniques are available for estimation of FD. However, different technique leads to different results and finding the accuracy of an algorithm for fractal dimension estimation is still a great challenge. In this comparative analysis the most well liked methods like differential box counting (DBC), relative DBC (RDBC), improved box counting (IBC), improved DBC (IDBC) of estimating FD of gray scale images are analysed. The analysis are performed over two sets of data base images called Brodatz texture data base images and FKP database and two set of generated synthetic texture images and another forty generated shrunken images. The merits and pitfalls of individual algorithm are detailed discussed. The outcomes of fitting error, recognition accuracy of each algorithm are figure out and properly outlined. This research analysis indicates that the precise selection of FD technique is essential for accurate estimation of roughness of specific objects.

Keywords FD · DBC · RDBC · IBC · IDBC · ITBC

1 Introduction

Fractal dimension (FD) is a term used in fractal geometry to evaluate surface roughness of complex objects found in nature like cloud, mountain and coastlines. Generally utmost of the objects of the nature are convoluted and aberrant pattern, therefore it was very difficult to analysed and characterize these patterns by Euclidean geometry reported in [1, 2]. In order to describe these complex objects, fractal dimension comes into existence and it was initially presented by Mandelbrot [3]. Now a days fractal dimension becomes most popular in many kind of application such as pattern recognition, texture analysis, medical signal analysis and image segmentation reported in [4]. Many researchers contributed their effort in field of fractal geometry. Thus, different technique has been presented to estimate fractal dimension. Voss described and partitioned these techniques into three key concepts such as box counting, variance and spectral method reported in [5]. The box counting was most successful and widely used technique for estimating FD in various field of application due to its simplicity and easy implementation [6]. In this regard many box counting and its improved version comes into existence and found in many literatures [7–14]. Sarkar and Chaudhuri [8] proposed most appropriate technique, called the differential box-counting (DBC) in digital domain for FD estimation by taking maximum and minimum intensity point described in many literatures [15–21]. Jin et al. [10] presented relative DBC by adopting a convenient algorithm in terms of higher and poorer confines for better computation. Biswas et al. [22] presented the modified version of DBC by taking a parallel algorithm for efficient estimation. Chen et al. [1] presented another approach as similar to RDBC

✉ Soumya Ranjan Nayak
nayak.soumya17@gmail.com

Jibitesh Mishra
jmishra@cet.edu.in

Pyari mohan Jena
mohanjena.cse@gmail.com

1 Department of Information Technology, College of Engineering and Technology, Bhubaneswar, India

2 Department of Computer Science and Application, College of Engineering and Technology, Bhubaneswar, India

3 Department of Computer Science and Engineering, College of Engineering and Technology, Bhubaneswar, India

called shifting DBC (SDBC) by using the concept of shift operation. The improved Box counting (IBC) algorithm was presented by Li et al. [11] in terms of three modifications such as height selection, exact box number estimation and surface partitioning. Liu et al. [12] was presented another improved version of DBC approach called improved DBC (IDBC) by adopting three concept such as customize box-counting concept, choosing correct size of grid and shifting the box in spatial coordinates for better FD estimation. In order to enhance the accuracy over conventional box counting, Kaewaramsri et al. [13] was presented triangle box-counting (TBC) technique by partitioning the grid into two equal triangles for more accurate estimation of FD for binary images. Recently, an improved TBC (ITBC) was presented by same author [14] by dividing the each grid box into two triangles in terms of four patterns such as upper and lower right diagonal, upper and lower left diagonal for more precision box count. Recently Nayak et al. [23] presented modified triangle box counting technique that provides optimize performance in terms of less fitting error and simultaneously solve both under counting and over counting problem by implementing asymmetric triangle box partition.

The remainder of this article are represented as follows: in Sect. 2, the review of the background of fractal dimension and several DBC methods are discussed. In Sect. 3, the experimental results are outlined. At last our concluding remarks are outlined in Sect. 4.

2 Related background works

The FD is a major characteristic of fractal geometry to estimate surface roughness of whole image. The basic rules for estimation of fractal dimension of a whole image which is based upon the concept of self-similarity. When a large fractal object is divided into smaller parts each part is same as whole object. Consider a bounded set A in Euclidean n-space. The set is supposed to be self-similar if A is the union of $N_r(A)$ different non overlapping copies, each different copies are scaled down by a ratio r and every scaled down copies are look like to the original copies itself. While in this regard, many techniques have been projected to improve the accuracy of FD estimation. The following subsections describe existing well liked methods which we have taken into consideration for our comparisons purpose and identify which technique is best suitable for estimation of FD of gray scale images. FD can be evaluated based on the concept of self-similarity. According to the box counting theorem [24], fractal dimension D of a set A can be evaluated as follows, FD can be estimated from the least square regression line of $\log N_r(A)$ verses $\log(1/r)$.

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

2.1 Differential box counting

Sarkar and Chaudhuri [8] projected the differential box-counting (DBC) method for evaluation of FD of gray-scale images. In order to implement this algorithm, they represent gray-scale image in $3D$ space, where $2D$ space like (x, y) represents an image plane and third coordinates like z represents the gray level. For this experimental analysis, they took a square image of size $M \times M$ and partitioning into $L \times L$ grids. Each and every grid comprise a stake of boxes of size is $L \times L \times H$, where H indicates the height of a 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, then the box-count $n_r(i,j)$ can be evaluated as follows:

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

By taking involvement from all boxes, $N_r(A)$ is counting for different value of L as follows:

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

2.2 Relative differential box counting

Based on original DBC, Jin et al. [10] 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 upper and lower limits of scale ranges for accurately estimation of fractal dimension of texture images. These upper and lower scale ranges are determined from size of the image. Finally $N_r(A)$ can be calculated as follows:

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

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

2.3 Improved box counting

In similar to DBC and RDBC, Li et al. [11] presented another improved BC (IBC) mechanism by adopting three major parameters like height selection, exact box number estimation and surface partitioning. They are selecting box height by using the formula as below

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

where a is a positive integer and set the appropriate value a as 3 (as the author [11] suggested a as 3 for accurate estimation of scaling factor) and σ represents standard

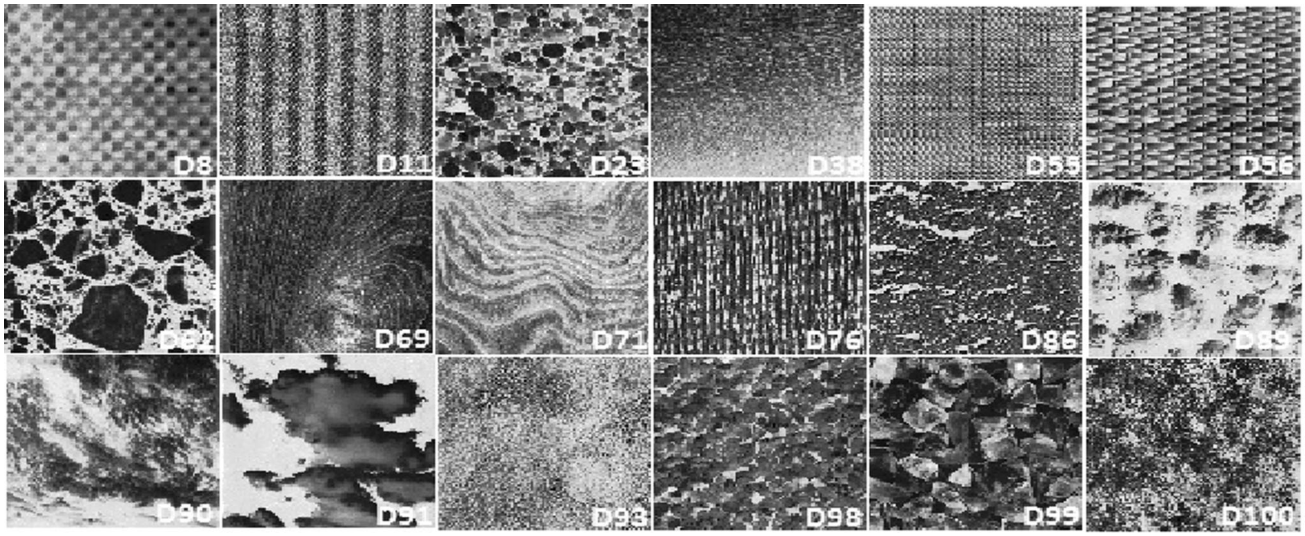

Fig. 1 Sixteen Brodatz database texture images

deviation and $2a\sigma$ represents image roughness. Finally $n_r(i,j)$ can be evaluated as follows:

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

Final $N_r(A)$ can be evaluated based on the Eq. (3).

2.4 Improved differential box counting

Liu et al. [12] proposed another improved version of DBC called improved DBC (IDBC) for estimating FD of gray scale image. In their proposed method, three modification have been done such as customize box-counting concept, choosing correct size of grid and shifting the box in spatial coordinates and $n_r(i,\ j)$ calculated by taking maximum contribution from original and shifted grid all grid as follows in Eq. (7).

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

Inline to DBC and IBC $N_r(A)$ can be computed based on the Eq. (3).

2.5 Improved triangle box counting

Kaewaramsri et al. [14] was presented an improved triangle box counting (ITBC) for accurate estimation of FD in digital domain by implementing equally triangle box partition. As a result, there are two patterns, p1 and p2, such that p1 is comprises of upper and lower right diagonals while p2 is comprises of upper and lower left diagonals. Then, both p1 and p2 are computed by averaging box-counts from both patterns, respectively. Finally $n_r(i,j)$ of each grid box can be calculated by taking maximum contribution from both patterns as Eq. (8).

$$n_r(i,j) = Max(p_1, p_2) \tag{8}$$

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

3 Result and discussion

In this section, experimental result and discussion are done to evaluate the performance of the five chosen methods and pointed out which method provides best suitable result in terms of less fit error and simultaneously solve all kinds of problems like over counting, under-counting; and also indicate which algorithm gives better recognition accuracy. The experiments are carried out on a system with a matlab14(a) in windows 8, 64 bit operating system, Intel (R) i7-4770 CPU @ 3.40 GHz. In this experimental analysis, we have considering five well liked methods such as DBC, RDBC, IBC, IDBC and ITBC and finally compared through five experiments, which have a set of original sixteen real Brodatz images [25] represented in Fig. 1, one set of twelve synthetic images represented in Fig. 4, and one set of eighteen generated synthetic texture like images represented in Fig. 6, one set of FKP database images [26] and one set of generated shrunken images. The details description of each experimental analysis is discuss in following subsections.

3.1 Test on Brodatz images

In this section, we are using a set of 16 real texture images [25] of size 256 × 256 from Brodatz database for our experimental analysis which is represented in Fig. 1, However, fractal dimension can be calculated using linear fit straight line verses $\log N_r(A)$ and $\log(1/r)$. Then the error fit can be estimated from the root mean square distance of the data points from the line by using Eq. (9). Let $y = mx + c$ be the fitted straight line, where x axis represents $\log N_r(A)$ and y axis represents $\log(1/r)$. Their corresponding FD and error fit are listed on Tables 1 and 2, The FD generated from DBC method are in the range from 2.309 to 2.697, similarly the other measure like RDBC, IBC, IDBC, and ITBC method are range from 2.291 to 2.712 and 2.308 to 2.729 and 2.353 to 2.745 and 2.272 to 2.659 respectively are listed in Table 1, and individual error fit of Brodatz images using five methods are listed in Table 2. The average error fit is estimated from each methods are 0.0626, 0.0611, 0.0622, 0.0531 and 0.0624 respectively are listed in Table 3 and presented on Fig. 2. The lower error fit indicates higher accuracy. We have seen from this experimental analysis, that only IDBC method yields less error fit for each individual images as compared to other four methods, that means it is crystal clear that IDBC method accurately estimates fractal dimension with less fit error because this method counted accurate number of boxes as compared to other existing method, hence resulted error fit is quit less as compared to other existing chosen methods and simultaneously provide less error fitting error in each individual images, whose data are presented in Fig. 3.

Table 1 Computational FD of 16 Brodatz database images

Image name	Fractal dimension				
	DBC	RDBC	IBC	IDBC	ITBC
D8	2.378	2.405	2.422	2.435	2.352
D11	2.672	2.684	2.704	2.715	2.623
D23	2.599	2.605	2.619	2.654	2.559
D38	2.538	2.609	2.615	2.592	2.507
D55	2.697	2.712	2.729	2.745	2.659
D56	2.605	2.607	2.617	2.669	2.565
D62	2.572	2.577	2.591	2.631	2.535
D69	2.495	2.524	2.546	2.546	2.455
D71	2.496	2.528	2.544	2.551	2.470
D76	2.691	2.695	2.705	2.745	2.656
D86	2.656	2.661	2.672	2.708	2.603
D89	2.510	2.539	2.542	2.572	2.463
D90	2.450	2.480	2.483	2.500	2.423
D91	2.309	2.291	2.308	2.353	2.272
D93	2.611	2.649	2.656	2.657	2.566
D98	2.520	2.520	2.533	2.571	2.471
D99	2.494	2.492	2.507	2.546	2.452
D100	2.682	2.690	2.703	2.727	2.635

Table 2 Computational error fit of 16 Brodatz database images

Image name	Error fit				
	DBC	RDBC	IBC	IDBC	ITBC
a8	0.071	0.073	0.074	0.065	0.071
a11	0.056	0.053	0.055	0.046	0.057
a23	0.069	0.067	0.068	0.060	0.069
a38	0.045	0.048	0.048	0.035	0.045
a55	0.052	0.049	0.051	0.041	0.053
a56	0.069	0.068	0.068	0.059	0.069
a62	0.071	0.068	0.069	0.060	0.070
a69	0.054	0.052	0.054	0.042	0.051
a71	0.059	0.059	0.060	0.051	0.059
a89	0.060	0.057	0.058	0.050	0.060
a90	0.064	0.061	0.062	0.054	0.065
a91	0.070	0.069	0.069	0.059	0.069
a93	0.068	0.067	0.068	0.059	0.065
a98	0.076	0.074	0.076	0.069	0.075
a99	0.049	0.046	0.047	0.039	0.051
a100	0.070	0.068	0.069	0.061	0.069

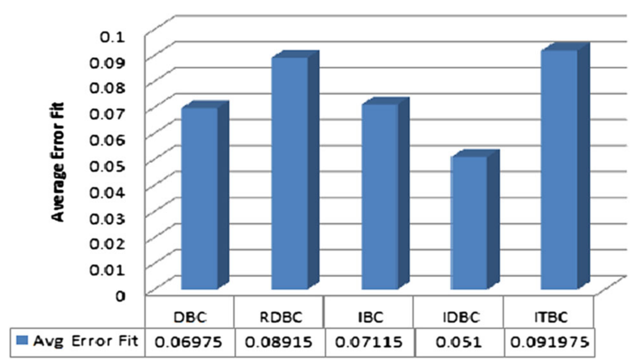

Fig. 2 Average fitting error of each corresponding method

$$Error\ fit = \frac{1}{n}\sqrt{\sum_{i=1}^{n}\frac{(mx_i + c - y_i)}{1 + d^2}} \tag{9}$$

3.2 Test on synthetic image 1

In this experimental setup, a set of 12 synthesize images are generated, which are presented in Fig. 4, In this process, each image are generated from original image by incrementing each intensity point (by formula $2 \times (k - 2)$,

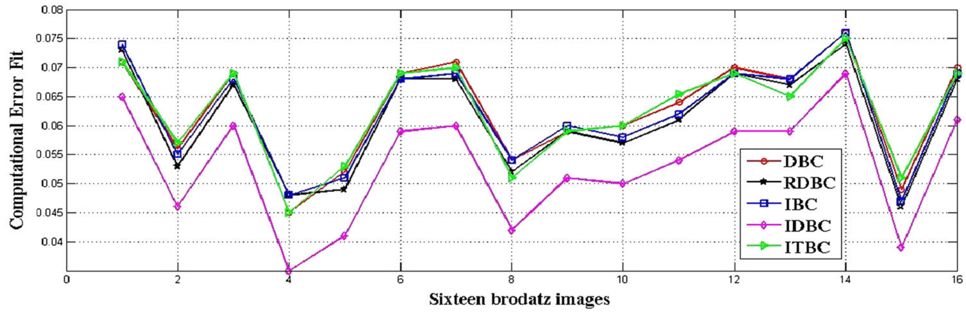

Fig. 3 Computational error fit of Sixteen Brodatz database texture images

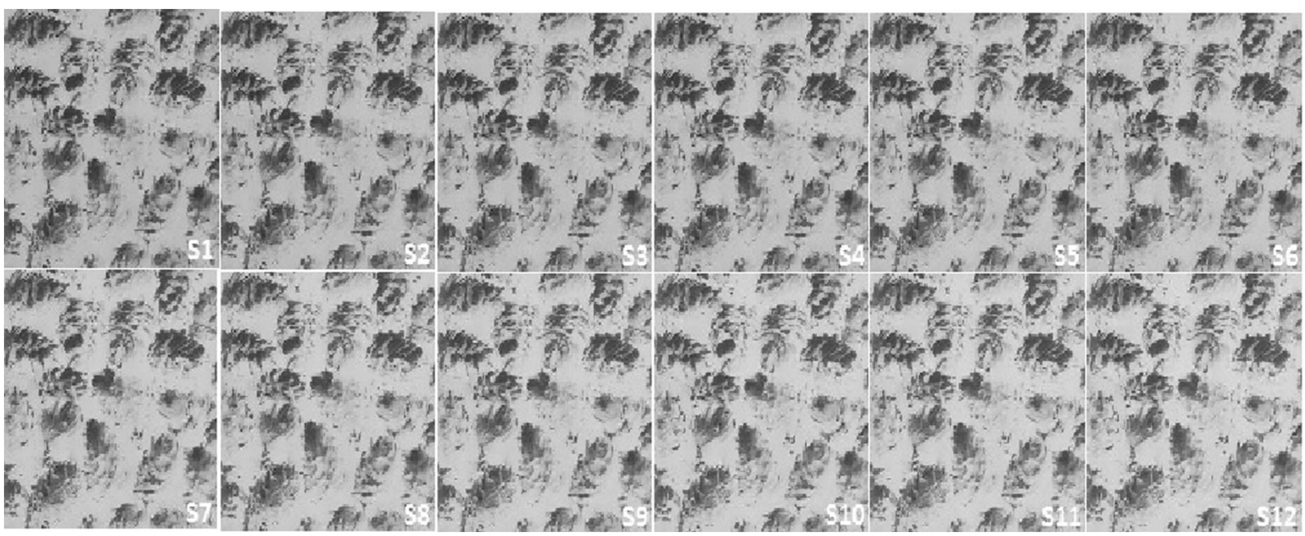

Fig. 4 Twelve generated synthetic texture images

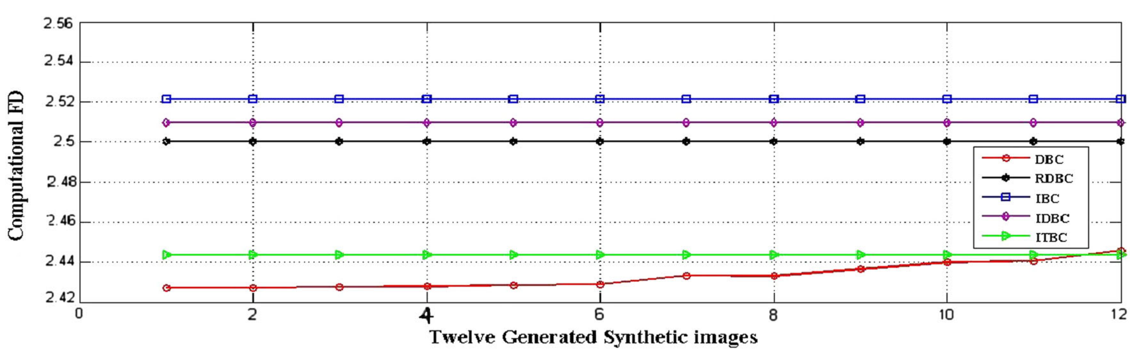

Fig. 5 Computational FD of Twelve generated texture images

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, therefore alternatively we can say that all the intensity surface are only shifted up to some gray levels from the original one in z direction. Theoretically, it was clear that, if we either increasing or decreasing by a constant value, then the FD should stay same as original one because theoretically both have the equal degree of roughness. Figure 5 represents computational FD and we have seen that expect DBC all other methods provide same estimated value because in DBC, gray level is fixed on the certain place in the z direction; the first box is in zero gray level and last box is in 255 gray level, this may cause over counting. From this result, we have crystal clear that except DBC method all other methods solve over-counting problems.

3.3 Test on synthetic image 2

In this experimental work, we have created 18 texture like images, which are represented in Fig. 6. From this experimental analysis, it is crystal clear that the except IDBC, all other chosen methods are not able to capture the roughness of these kinds of images having sharp gray level absorption at border of the box of block because the boxes with fix size at the fix location in spatial coordinate and some of the boxes will miss when sharp gray level absorption exist that leads to under counting problem. That means IDBC able to solve both kind of problem like over-counting and under-counting simultaneously. Figure 7 represents about the graphical presentation of computational FD of eighteen generated images. The details algorithm of generated synthetic images is listed below:

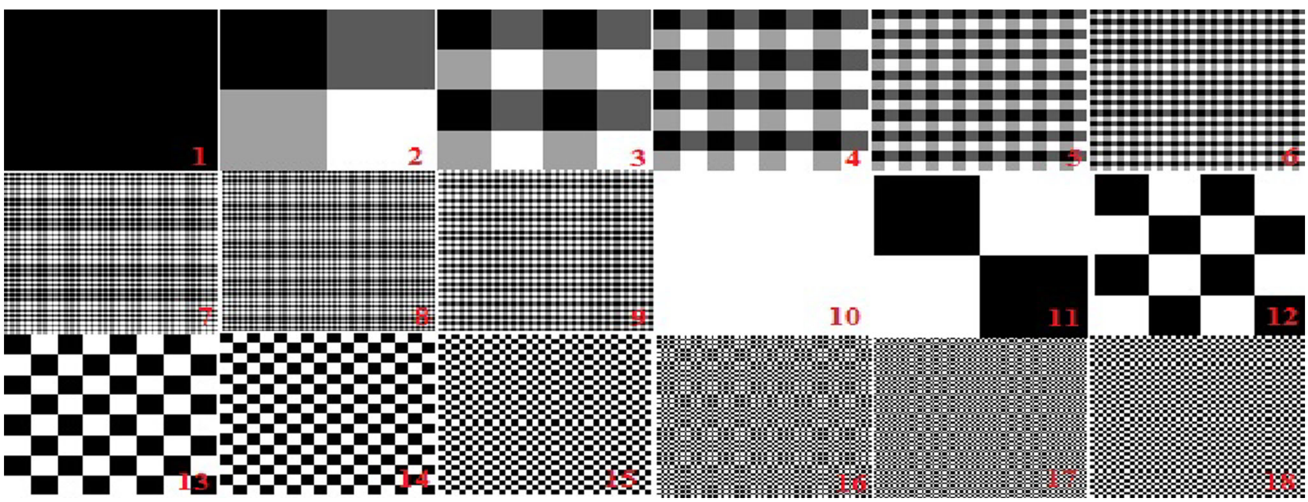

Fig. 6 Eighteen generated synthetic texture like images

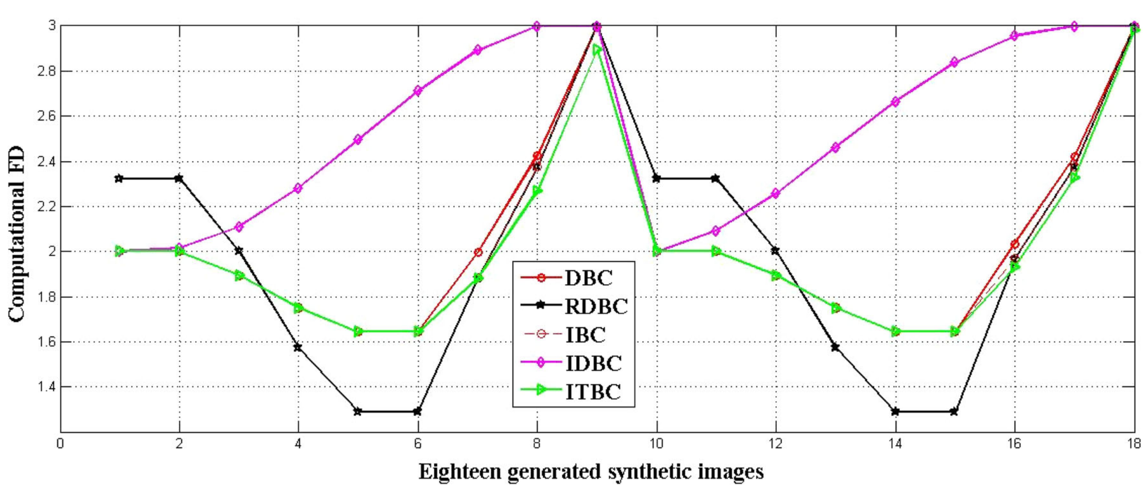

Fig. 7 Computational FD of Eighteen texture like images

```
Algorithm:1

The algorithm process to generate eighteen synthetic images

Start
Create an image B of size 256*256 and set zeros to all pixels
[x, y]=Size (B)
Create an array s=[x/2……….1] % Create array names S for down sampling.
For i<=length(s)
a=[1: s(p): x];
For i<=length (a), j<=length (a)
Flag=0;
if mod(i,2)==1 && Flag==0
B(a(i):a(i)+s(p)-1,a(j):a(j)+s(p)-1)=0;
Flag=Flag+1;
elseif mod(i,2)==1 && Flag==1
B(a(i):a(i)+s(p)-1,a(j):a(j)+s(p)-1)=90;
Flag=Flag-1;
elseif mod(i,2)==0 && Flag==0
B(a(i):a(i)+s(p)-1,a(j):a(j)+s(p)-1)=160;
Flag=Flag+1;
elseif mod(i,2)==0 && Flag==1
B(a(i):a(i)+s(p)-1,a(j):a(j)+s(p)-1)=255;
Flag=Flag-1;
End For
End For
Stop
```

3.4 Test on FKP database images

In this section, we have used polyuFKP data base images [26] to evaluate the recognition accuracy of each selected method. Feature extraction provides a unique way to identify the object images and extracted its feature values. In this experimental setup, we have performed feature extraction technique on individual FD technique, which we have taken into consideration. This data base comprises of 990 numbers of images of four classes like left index (LI), right index (RI), left middle (LM) and right middle (RM). Sample images are presented in Fig. 8. Each individual class contains nine single figures with image resolution of 64 × 64 pixels. This analysis demonstrates the recognition rate of each chosen methods in terms of fractal feature.

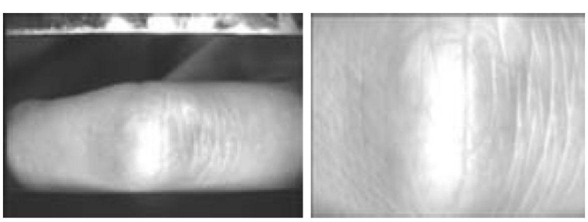

Fig. 8 **a** An acquired FKP image, **b** an ROI image with 110 × 220 pixels

Table 3 Average error fit of individual dataset by using different methods

Images	Average error fit				
	DBC	RDBC	IBC	IDBC	ITBC
Brodatz	0.0626	0.0611	0.0622	0.0531	0.0624
Synthetic 1	0.0641	0.0671	0.0687	0.0563	0.0640
Synthetic 2	0.0859	0.1636	0.0882	0.0382	0.0934
Shrunken images	0.0664	0.0648	0.0655	0.0564	0.1481
Avg error fit	0.06975	0.08915	0.07115	0.051	0.091975

Table 4 The coefficient variances of FD of Forty shrunken images

Image name	Coefficient of variation				
	DBC	RDBC	IBC	IDBC	ITBC
D1	0.0074	0.0066	0.0037	0.0071	0.0116
D3	0.0795	0.0826	0.0746	0.0768	0.0676
D4	0.0448	0.0421	0.0346	0.0356	0.0435
D5	0.0184	0.0196	0.0145	0.0146	0.0173
D6	0.0752	0.1009	0.0698	0.1133	0.1119
D8	0.0163	0.0157	0.0135	0.0098	0.0052
D9	0.0508	0.0451	0.0413	0.0602	0.0624
D11	0.0417	0.0413	0.0343	0.0402	0.0522
D22	0.0456	0.0402	0.0402	0.0436	0.0365
D23	0.0072	0.0086	0.0074	0.0084	0.0095
D24	0.0514	0.0457	0.0387	0.0392	0.0465
D26	0.0291	0.0184	0.0131	0.0248	0.0361
D27	0.0080	0.0091	0.0071	0.0059	0.0120
D28	0.0241	0.0164	0.0158	0.0251	0.0302
D29	0.0345	0.0348	0.0264	0.0296	0.0336
D30	0.0080	0.0105	0.0140	0.0218	0.0192
D31	0.0070	0.0083	0.0115	0.0189	0.0164
D32	0.0285	0.0158	0.0109	0.0414	0.0415
D33	0.0226	0.0136	0.0110	0.0228	0.0237
D34	0.0407	0.0460	0.0325	0.0635	0.0619
D38	0.0698	0.0834	0.0727	0.0823	0.0886
D52	0.0583	0.0571	0.0431	0.0504	0.0520
D53	0.0831	0.0724	0.0732	0.0893	0.0921
D55	0.0561	0.0442	0.0423	0.0519	0.0527
D56	0.0025	0.0025	0.0059	0.0050	0.0049
D57	0.0428	0.0292	0.0288	0.0380	0.0444
D65	0.0065	0.0039	0.0072	0.0095	0.0075
D66	0.0230	0.0222	0.0155	0.0239	0.0270
D68	0.0437	0.0458	0.0430	0.0459	0.0421
D69	0.0459	0.0520	0.0463	0.0473	0.0501
D76	0.0304	0.0233	0.0229	0.0389	0.0432
D77	0.1068	0.0935	0.0884	0.0996	0.1044
D78	0.0700	0.0681	0.0565	0.0590	0.0650
D79	0.0705	0.0612	0.0610	0.0630	0.0675
D80	0.0501	0.0517	0.0445	0.0634	0.0485
D82	0.0443	0.0279	0.0286	0.0400	0.0456
D84	0.0398	0.0297	0.0291	0.0317	0.0273
D92	0.0254	0.0254	0.0161	0.0248	0.0328
D95	0.0054	0.0059	0.0080	0.0219	0.0207
D111	0.0185	0.0160	0.0140	0.0234	0.0287

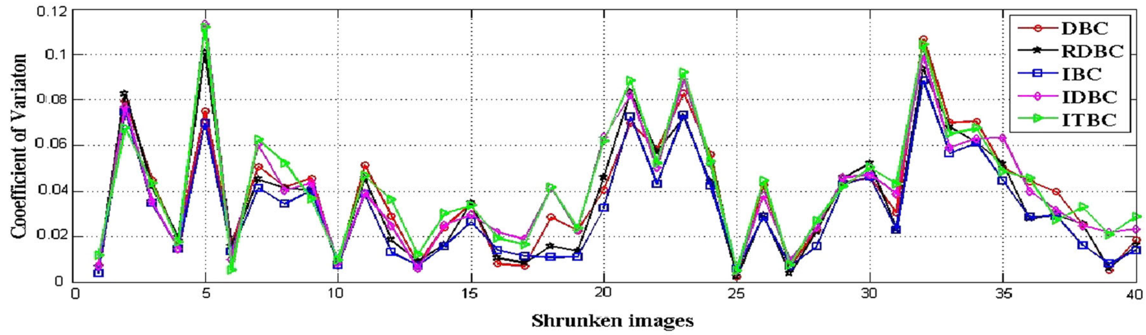

Fig. 9 The comparison diagram of coefficient variation of 40 shrunken images

Feature extraction technique consists of two phases namely training and testing phase. The training feature was extracted by different methods with different reduction factor of size 2, 4, 8, 16, 32 as per the box-counting mechanism on left index, left middle, right index and right middle image set. We extracted 5 feature row of each sample, therefore $990 \times 4 = 3960$ total number of rows for each class in order to classify the box counting. Therefore 2, 4, 8, 16, 32 box size (scaling factor), and the total number of rows required 3960×5 for 5 classes. Then this extracted feature are applied to random forest algorithm [27] for classification because random forest is one the best accepted technique for classification in machine learning, which provides high accuracy of result in case recognition over all other approaches. It can use for classification over large database to achieve high accuracy [27] and it follows tree pattern values of the random class nodes and the level of distribution. The highest accuracy of FKP database image is 99.94 obtained from IBC method, while other methods like DBC, RDBC, IDBC and ITBC are obtained as 94.36, 94.41, 96.26 and 97.46 respectively. Hence, the more accurate FD feature lead to the higher accuracy of FKP recognition.

3.5 Test on shrunken images

In this experimental setup, we took forty images from Brodatz database of sizes 640×640 and reduce its size into four different sizes like 512×512, 256×256, 128×128, and 64×64 to observe the coefficient variances of fractal dimension. Since the reduced images are generated from the same image having similar texture then the coefficient variance should yields less. The less variance in one group indicates better accuracy as it is reduced from same image. The coefficient variance results of fractal dimension are listed in Table 4. The comparison diagram is presented as shown in Fig. 9. The IBC method coefficient variance in groups which are inferior to the other method are 24, which means 60% of the results are superior to the other methods and the average coefficient variance of improved proposed method are quit less than other five chosen methods. The average coefficient variance of DBC, RDBC, IBC, IDBC, and ITBC evaluated as 0.0383, 0.0359, 0.0315, 0.0402, and 0.0420 respectively. From this experimental analysis, we have conclude that IBC algorithm have better performance in identifying the same class of images with different scales compared with other five competent methods.

4 Conclusion

In this paper, we have tried to made comparative study on different algorithm for estimation of fractal dimension of gray scale images in terms of five experimental analyses. Our findings reveal that the IDBC method, however yields a more accurate estimation in terms of less fitting error and solve both over counting and under counting problems simultaneously and also it was able to estimating accurate fractal dimension if sharp gray level abruption just at border of two neighbouring box blocks. Whereas the other methods like RDBC, IBC and ITBC methods only able to solve over counting problem, while DBC method won't able to solve both problems. However ITBC method performs better for recognition accuracy for classification as compare to other chosen methods. This study demonstrates that a careful selection of fractal dimension algorithm is required for specific objects. Further systematic validation is needed on more kinds of surface generation algorithm to analysis fractal dimension on specific objects.

References

1. Chen WS, Yuan SY, Hsieh CM (2003) Two algorithms to estimate fractal dimension of gray-level images. Opt Eng 42:2452–2464
2. Asvestas P, Matsopoulos GK, Nikita KS (1998) A power differentiation method of fractal dimension estimation for 2-D signals. J Vis Commun Image Represent 9:392–400
3. Mandelbrot BB (1982) The fractal geometry of nature. Freeman, San Francisco

4. Lopes R, Betrouni N (2009) Fractal and multifractal analysis: a review. Med Image Anal 13:634–649
5. Balghonaim AS, Keller JM (1998) A maximum likelihood estimate for two-variable fractal surface. IEEE Trans Image Process 7:1746–1753
6. Peitgen HO, Jurgens H, Saupe D (1992) Chaos and fractals: new frontiers of science, 1st edn. Springer, Berlin
7. Gangepain J, Roques-Carmes C (1986) Fractal approach to two dimensional and three dimensional surface roughness. Wear 109:119–126
8. Sarkar N, Chaudhuri BB (1994) An efficient differential box-counting approach to compute fractal dimension of image. IEEE Trans Syst Man Cybern 24:115–120
9. Buczkowski S, Kyriacos S, Nekka F, Cartilier L (1998) The modified box-counting method: analysis of some characteristics parameters. Pattern Recognit 3:411–418
10. Jin XC, Ong SH, Jayasooriah (1995) A practical method for estimating fractal dimension. Pattern Recognit Lett 16:457–464
11. Li J, Du Q, Sun C (2009) An improved box-counting method for image fractal dimension estimation. Pattern Recognit 42:2460–2469
12. Liu Y, Chen L, Wang H, Jiang L, Zhang Y, Zhao J, Wang D, Zhao Y, Song Y (2014) An improved differential box-counting method to estimate fractal dimensions of gray-level images. J Vis Commun Image Represent 25:1102–1111
13. Woraratpanya K, Kakanopas D, Varakulsiripunth R (2012) Triangle-box counting method for fractal dimension estimation. ASEAN Eng J Part D 1:5–16
14. Kaewaramsri Y, Woraratpanya K (2015) Improved triangle box counting method for fractal dimension estimation. Ic2It2015 361:53–61
15. Wenlu X, Weixin X (1997) Fractal-based analysis, of time series data and features extraction. Signal Process 13:98–104
16. Yu L, Zhang D, Wang K, Yang W (2005) Coarse iris classification using box-counting to estimate fractal dimensions. Pattern Recognit 38:1791–1798
17. Nayak SR, Ranganath A, Mishra J (2015) Analysing fractal dimension of color images. In: International conference on computational intelligence and networks (CINE). IEEE, pp 156–159
18. Nayak SR, Mishra J (2016) An improved method to estimate the fractal dimension of colour images. Perspect Sci 8:412–416
19. Nayak SR, Mishra J (2016) An improved algorithm to estimate the fractal dimension of gray scale images. In: Signal processing, communication, power and embedded system (SCOPES), international conference on IEEE, pp 1109–1114
20. Nayak SR, Mishra J, Padhi R (2016) On estimation fractal dimension of gray scale images using modified differential box counting method. Int J Comput Sci Inf Secur 14:764–772
21. Nayak SR, Mishra J (2017) On estimation of fractal dimension of noisy images. Indian J Sci Technol 10:1–6
22. Biswas MK, Ghose T, Guha S, Biswas PK (1998) Fractal dimension estimation for texture images: a parallel approach. Pattern Recognit Lett 19:309–313
23. Nayak SR, Mishra J (2017) A modified triangle box-counting with precision in error fit. J Inf Optim Sci. https://doi.org/10.1080/02522667.2017.1372155
24. Barnsley MF (1993) Fractal everywhere, 2nd edn. Academic Press Professional, New York
25. Brodatz P (1966) Texture: a photographic album for artists and designers. Dover Publications, New York
26. Zhang L, Zhang L, Zhang D (2009) Finger-knuckle-print: a new biometric identifier. In: IEEE international conference on image processing, pp 1981–1984
27. Xu B, Ye Y, Nie L (2012) An improved random forest classifier for image classification. In: International conference on information and automation (ICIA), 23 July 2012