Pattern Recognition 42 (2009) 2460 – 2469

Contents lists available at ScienceDirect

Pattern Recognition

journal homepage: www.elsevier.com/locate/pr

An improved box-counting method for image fractal dimension estimation

Jian Li[a], Qian Du[b,*], Caixin Sun[a]

[a]State Key Laboratory of Power Transmission Equipment & System Security and New Technology, College of Electrical Engineering, Chongqing University, Chongqing 400044, China
[b]Department of Electrical and Computer Engineering, Mississippi State University, MS 39762, USA

ARTICLE INFO

Article history:
Received 6 September 2007
Received in revised form 14 January 2009
Accepted 2 March 2009

Keywords:
Fractal dimension
Box-counting dimension
Fractional Brownian motion
Texture image
Remote sensing image

ABSTRACT

Fractal dimension (FD) is a useful feature for texture segmentation, shape classification, and graphic analysis in many fields. The box-counting approach is one of the frequently used techniques to estimate the FD of an image. This paper presents an efficient box-counting-based method for the improvement of FD estimation accuracy. A new model is proposed to assign the smallest number of boxes to cover the entire image surface at each selected scale as required, thereby yielding more accurate estimates. The experiments using synthesized fractional Brownian motion images, real texture images, and remote sensing images demonstrate this new method can outperform the well-known differential boxing-counting (DBC) method.

© 2009 Elsevier Ltd. All rights reserved.

1. Introduction

Fractal geometry provides a mathematical model for many complex objects found in nature [1–3], such as coastlines, mountains, and clouds. These objects are too complex to possess characteristic sizes and to be described by traditional Euclidean geometry. Self-similarity is an essential property of fractal in nature and may be quantified by a fractal dimension (FD). The FD has been applied in texture analysis and segmentation [4–6], shape measurement and classification [7], image and graphic analysis in other fields [8,9].

There are quite a few definitions of FDs making sense in certain situations. Thus, different methods have been proposed to estimate the FD. Voss summarized and classified these methods into three major categories: the box-counting methods, the variance methods, and the spectral methods [10]. The box-counting dimension is the most frequently used for measurements in various application fields. The reason for its dominance lies in its simplicity and automatic computability [3]. Several practical box-counting methods were brought forward for FD estimation [11–17]. In Ref. [12], the differential box-counting (DBC) method was compared with other four methods proposed by Gangepain and Roques-Carmes [11], Peleg [18], Pentland [1], and Keller [19], respectively. The DBC method was considered as a better method, as was also supported by the investigation conducted in Refs. [20,21].

However, in [14] the drawbacks of the DBC method were pointed out, such as the proneness of overcounting or undercounting the number of boxes, and then the shifting DBC and the scanning DBC were proposed which may provide more accurate estimates. Buczkowski et al. [13] also presented a new procedure to eliminate another two problems of a box-counting method, i.e., the boarder effect and non-integer values of selected copy scales.

According to our experience on the analysis of gray level images, the DBC method may provide unreasonable FDs of images, which may be less than two, particularly when these images are quite smooth. Therefore, we develop a new method for more accurate estimates, resulting from completely different strategies in box scale selection, box number determination, and image intensity surface partition. The experimental results using the synthesized fractional Brownian motion (fBm) images, real texture images, and remote sensing images, demonstrated its advantages.

This paper is organized as follows. In Section 2, the basic concept of FD and procedures of the original DBC method are introduced. The problems of the DBC method are discussed in this section as well. In Section 3, the new approaches for selecting appropriate box scale, determining box numbers, and partitioning image intensity surface are presented. In Section 4, fBm images, real texture images, smoothened texture images, and remote sensing images, are used to evaluate the proposed box-counting estimation. Section 5 draws the conclusions.

2. Basic definition and DBC estimate

The basic principle to estimate FD is based on the concept of self-similarity. The FD of a bounded set A in Euclidean n-space is defined as

$$D = \lim_{r \to 0} \frac{\log(N_r)}{\log(1/r)} \tag{1}$$

* Corresponding author.
E-mail address: du@ece.msstate.edu (Q. Du).

0031-3203/$ - see front matter © 2009 Elsevier Ltd. All rights reserved.
doi:10.1016/j.patcog.2009.03.001

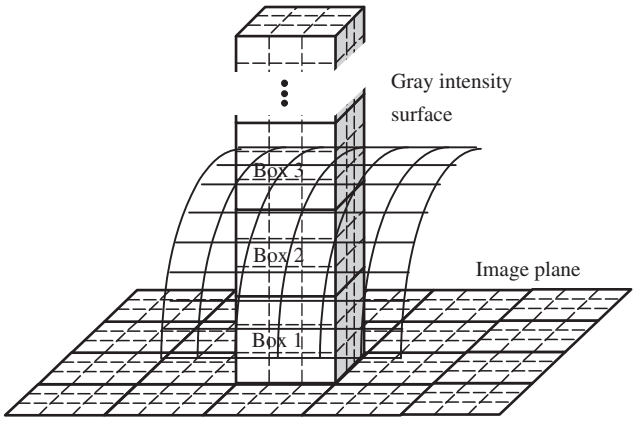

Fig. 1. Sketch of determination of the number of boxes by the DBC method.

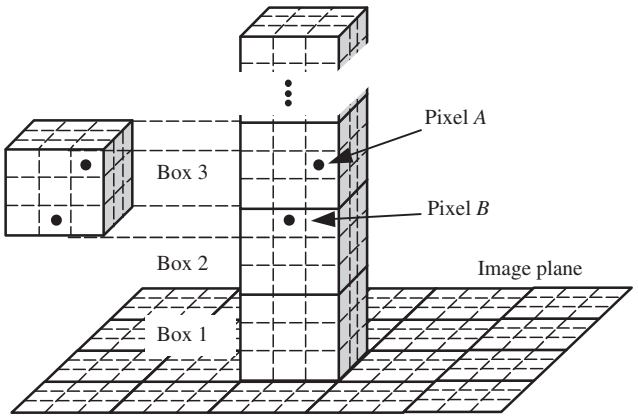

Fig. 2. Two pixels fall into two boxes with the size 3×3×3 but have the distance smaller than 3 in height direction.

where N_r is the least number of distinct copies of A in the scale r. The union of N_r distinct copies must cover the set A completely.

The FD can only be calculated for deterministic fractals. For an object with deterministic self-similarity, its FD is equal to its box-counting dimension D_B. However, natural scenes are not the ideal and deterministic fractals. The box-counting dimension D_B is an estimate of D, calculated by using a box-counting method. The DBC method is introduced as follows.

Consider an image of size $M\times M$ as a three-dimensional (3-D) spatial surface with (x, y) denoting pixel position on the image plane, and the third coordinate (z) denoting pixel gray level. In the DBC method, the (x, y) plane is partitioned into non-overlapping blocks of size $s\times s$. The scale of each block is $r=s$, where $M/2 \geqslant s > 1$ and s is an integer. N_r is counted in the DBC method using the following procedure. On each block there is a column of boxes of size $s\times s\times s'$, where s' is the height of each box, $G/s' = M/s$, and G is the total number of gray levels. For example, $s = s' = 3$ in Fig. 1 [12]. Assign numbers 1, 2, …, to the boxes as shown in Fig. 1. Let the minimum and maximum gray level in the (i, j)th block fall into the kth and lth boxes, respectively. The boxes covering this block are counted in the number as

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

where the subscript r denotes the result using the scale r. For example, $n_r(i, j) = 3-1+1$ as illustrated in Fig. 1. Considering contributions from all blocks, N_r is counted for different values of r as

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

Then the FD can be estimated from the least squares linear fit of log (N_r) versus log $(1/r)$.

There are three major problems in the aforementioned procedure in the original DBC method:

(1) *Box height selection*: When the boxes in the size $s\times s\times s'$ are assigned to cover the image surface, the value of s is limited to the image size but s' may be flexible. Intuitively, a smaller measurement error results from a smaller measurement scale. If the height of boxes is selected as $s' = sG/M$ as in the DBC method, it may have a larger value when s is increased. Then the box in a larger scale may result in greater computational error when counting the box numbers. How to select an appropriate box scale in height is the first problem to be resolved.

(2) *Box number calculation*: The DBC method assigns a column of boxes on a block starting from the gray level zero, which induces the second problem illustrated in Fig. 2, where pixels A and B represent the maximum and minimum gray levels of the block (they have the z coordinates corresponding to the gray level values). If a column of boxes with the size 3×3×3 cover this block, the two pixels are assigned in boxes 2 and 3 in Fig. 2. However, the distance between the two pixels in the z direction is smaller than 3. Only one box can cover this block even though its pixels with the maximum and minimum gray levels fall into two different boxes, which is also the result from Eq. (2). So the DBC method cannot acquire the least number of boxes that cover the block. A new approach to find the least numbers of boxes covering each block needs to be proposed.

(3) *Image intensity surface partition*: The pixels in a digital image are discretized results of a continuous image with infinite resolution. If the boxes are assigned as in the DBC method, there is a gap between any two spatially adjacent boxes. This indicates that all boxes in the DBC method cannot completely cover the entire image when the image intensity surface is considered to be continuous. Due to this fact, there must be estimation errors existing in the results from the DBC method. An approach to assign boxes that can cover the full image surface needs to be developed.

Due to the above three problems, the accuracy of the original DBC method is limited. Three modifications in our box-counting estimation method are proposed for improvement, which is presented in Section 3.

3. A new box-counting method

3.1. Basic concept of the new box-counting method

Let $z = f(x, y)$ be an ideal fractal image with a 3-D continuous surface. It is divided into distinct copies and a copy has a minimum denoted as $z_1 = f(x_1, y_1)$ and a maximum denoted as $z_2 = f(x_2, y_2)$, when $x_1 \leqslant x < x_2$ and $y_1 \leqslant y < y_2$. Let d_x and d_y be the lengths of the copy in the directions of x and y, respectively, and d_z be the height in the direction of z. Because $z = f(x, y)$ is continuous, we can obtain $d_x = x_2 - x_1$, $d_y = y_2 - y_1$, and $d_z = z_2 - z_1$. Suppose $d_x = d_y$ and define the box scale as $r = d_x = d_y$. Assume a column of boxes with the size of $d_x\times d_x\times d_x$ cover the copy of surface completely. The number of this column of boxes is equal to the integer part of $(d_z/r+1)$. Using Eq. (1), the FD of an ideal fractal can be determined via least squares linear fit. In other words, Eq. (1) indicates that there exists definitely a scale range $[r_1, r_2]$ in which the FD estimate of the ideal fractal surface can be calculated as

$$D = -\frac{\log N_{r_1} - \log N_{r_2}}{\log r_1 - \log r_2} \tag{4}$$

where N_{r_1} and N_{r_2} are the numbers of boxes covering the ideal fractal surface when the box scales are r_1 and r_2, respectively. When r_1 and r_2 approach zero, the scale range can be found such that the FD estimate error approaches zero. Any two values of r in $[r_1, r_2]$ and their corresponding N_r can be used to substitute the parameters in (4) and obtain D since N_r versus r is an exact straight line in a log–log plot in this scale range. The negative slope of the straight line is equal to D.

Due to the limited resolution, FD estimates of digital image surfaces have discrepancy from their true FDs, even though they are ideal fractals. If appropriate approaches for box number counting are employed, errors of FD estimates could be small enough. From the preceding description about FD calculation of an ideal and continuous fractal surface, the basic concepts of three modifications to be proposed for the DBC method are briefly summarized as below with more details being presented in the following sections.

(1) Let a new scale $r' = r/c$, where c is a positive real number greater than 1. The number of boxes covering a copy is calculated as the integer part of $(d_z/r'+1)$. Because $r' < r$ and $d_z/r' > d_z/r$, the residual part of d_z/r' is smaller than that of d_z/r. Thus, the errors introduced by the box scale r' are smaller than r in the original DBC method. This leads to the modification A in Section 3.2.
(2) d_z is used to count the numbers of box covering a copy of image surface instead of z, the gray levels of pixels, being used for Eq. (2) in the original DBS method, resulting in more accurate box number counting. This leads to the modification B in Section 3.3.
(3) The scale r for an ideal and continuous surface is defined as $r = d_x = d_y$, where $d_x = x_2 - x_1$ and $d_y = y_2 - y_1$. Therefore, for a digital image, if the distance between any two neighboring pixels are 1, then the scale $r = s-1$ rather than r in the original DBC method, when $s \times s$ pixels exist in a copy of a digital image surface. This leads to the modification C in Section 3.4.

3.2. Box height selection (modification A)

Let the mean and the standard deviation of a digital image be μ and σ. Suppose most pixels fall into the interval of gray level within $[\mu - a\sigma, \mu + a\sigma]$. The box height r' is selected as

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

where a is a positive integer and $2a\sigma$ can indicate image roughness. How to select the optimum value of a will be presented in Section 4.1. As a result, a box with smaller height is chosen for an image surface with higher intensity variation.

Compared to the height of boxes in the DBC method, the height of boxes in the improved DBC method is much smaller at different box scale of r. For example, if an image has the size of 128×128 with 128 gray levels and the standard deviation σ being 15, the box height is $r' = r = 3$ in the DBC method and it becomes $r' = r/91$ when $a = 3$ according to Eq. (4). So the new method uses finer scales to count the numbers of boxes covering each block and the entire image surface through automatic adjustment based on image smoothness.

3.3. Box number calculation (modification B)

If the maximum and minimum gray levels of the (i, j)th block are l and k, respectively, the number of boxes that cover the block surface can be calculated as

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

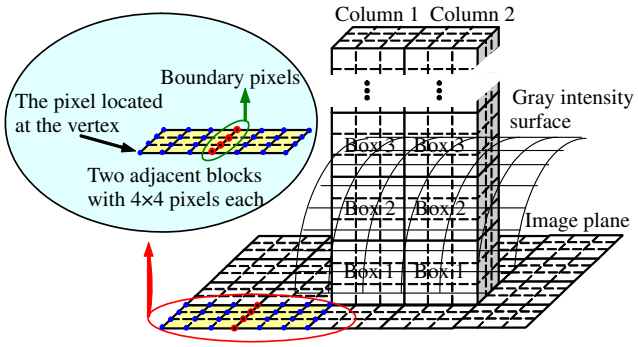

Fig. 3. Box number determination by the improved DBC method.

where ceil $(\cdot)$ denotes the function to round a value to the nearest and greater integer. The physical meaning of Eq. (5) is that the boxes are assigned from the minimum gray level of the block rather than gray level 0. It is expected that $n_r(i, j)$ is the least number of boxes covering the surface of the (i, j)th block.

For the example shown in Fig. 2, $r' = 3$ and the distance between the two pixels A and B in height direction is smaller than 3. By using Eq. (6), the number of boxes coving the block is $n_r = 1$, which is the exact number of boxes covering the block.

3.4. Image intensity surface partition (modification C)

To ensure that the image intensity surface is completely covered, the following partition scheme is applied:

(1) Partition an image with the size $M \times M$ into equivalent blocks with $s \times s$ pixels. Any two spatially adjacent blocks overlap at the boundary by one row (or column, i.e., s pixels). For instance, an image of size 16×13 pixels shown in Fig. 3 is partitioned into 20 blocks. Each block possesses 4×4 pixels. Any two adjacent blocks overlap by 4 pixels on their shared boundary.
(2) Let the scale r of a block with $s \times s$ pixels be $s-1$, which represents the maximum distance between any two pixels in a block on the same vertical or horizontal direction. For example, scale r of each block in Fig. 3 is equal to 3.

After this partition, pixels on the boundary of two neighboring blocks, falling into the (i, j)th and the $(i+1, j)$th grids, are taken into account for computing $n_r(i, j)$ and $n_r(i+1, j)$ as well. However, those boundary pixels give different contributions to $n_r(i, j)$ and $n_r(i+1, j)$. If their values represent the true values of pixels in the (i, j)th block, they are used for the computation of $n_r(i+1, j)$ as approximate values of boundary pixels. Fortunately, their values approach the true values of boundary pixels in the $(i+1, j)$th block since the image intensity surface is considered to be contiguous. Hence, an image plane is partitioned in the way that any two neighboring blocks approach zero in distance and all boxes cover the image intensity surface completely. In Section 4, it will be shown that any two neighboring blocks overlapping only one row of pixels on their boundary is the best selection for the image intensity surface partition.

3.5. Algorithm summary

The detailed algorithm of the improved DBC method can be summarized as follows:

(1) Divide the image into blocks of size $s \times s$. Any two adjacent blocks overlap by one row (and column) of pixels at the boundary.

Table 1
Computation complexity of the DBC and improved DBC methods.

DBC			Improved DBC		
Computation step	Computation operation		Computation step		Computation operation
Let $r=s$. Repeat while $0 \leqslant i < M/s$ and $0 \leqslant j < M/s$	$x_L = i\times s$	Multiplication: 4	Let $r=s-1$. Repeat while $0 \leqslant i < (M-)/(s-1)$ and $0 \leqslant j < (M-1)/(s-1)$	$x_L = (i-1)\times(s-1)$	
	$x_U = =(i+1)\times s-1$	Subtraction: 4		$x_U = i\times(s-1)$	Multiplication: 4
	$y_L = j\times s$			$y_L = (j-1)\times(s-1)$	Subtraction: 6
	$y_U = =(j+1)\times s-1$			$y_U = j\times(s-1)$	
	Find z_{max} and z_{min} in the block	Comparison: $2s^2$		Find z_{max} and z_{min} in the block	Comparison: $2s^2$
	$l = \text{ceil}\ (z_{max}/r')$	Division: 2		$n_r = \text{ceil}\ [(z_{max}-z_{min})/r')$	Comparison: 1
	$k = \text{ceil}\ (z_{min}/r')$	Addition: 3		*or*	Addition: 1
	$n_r = l-k+1$	Subtraction: 1		$n_r = 1$ if $z_{max} = z_{min}$	Division: 1
					Subtraction: 1
	$N_r = \Sigma n_r$	Addition: 1		$N_r = \Sigma n_r$	Addition: 1
Number of blocks is $(M/s)^2$ when blocks are in the size $s\times s$.			Number of blocks is $[(M-1)/(s-1)]^2$ when blocks are in the size $s\times s$.		
Comparison: $2M^2$			Comparison: $(2s^2+1)[(M-1)/(s-1)]^2$		
Addition: $4\times(M/s)^2$			Addition: $2\times[(M-1)/(s-1)]^2$		
Subtraction: $5\times(M/s)^2$			Subtraction: $7\times[(M-1)/(s-1)]^2$		
Multiplication: $4\times(M/s)^2$			Multiplication: $4\times[(M-1)/(s-1)]^2$		
Division: $2\times(M/s)^2$			Division: $[(M-1)/(s-1)]^2$		

(2) Assign a column of boxes with a scale of $r\times r\times r'$ starting the pixel with the minimum gray level in the block. r' is selected according to Eq. (5). The scale r of the boxes, corresponding to the scale in Eq. (1), is considered to be $r=s-1$.
(3) Determine $n_r(i,j)$ for each block by using Eq. (6).
(4) For different scale r, the total number of boxes N_r covering the full image surface is computed with Eq. (3).
(5) Plot the least squares linear fit of $\log(N_r)$ versus $\log(r)$. The estimated FD is equal to the negative slope of the fitted straight line.

Table 1 presents the complexity of computation steps in both the DBC and improved DBC methods. Given an image with size $M\times M$ and the box of scale $r\times r\times r'$, the image is divided into blocks with number equal to $(M/s)^2$ and $[(M-1)/(s-1)]^2$ for the DBC and improved DBC methods, respectively. Thus, for a block $z(x,y)$, where $x_L \leqslant x < x_U$ and $y_L \leqslant y < y_U$, the computational complexity in both methods is on the similar level.

4. Experiments

4.1. Experiment on simulated fractal Brownian motion images

In the simulation experiments, fBm surfaces were used. The fBm images were produced using the following:

$$B(x,y) = \sum_{k=0}^{M-1}\sum_{l=0}^{M-1}[R(k,l)\cdot(k^2+l^2)^{-(\alpha+1)/2}e^{2\pi i(kx+ly)/M}] \quad (7)$$

where B is an fBm image of size $M\times M$ (say, $M=256$) and x and y are coordinates of a pixel from 0 to $M-1$. Here, R is a conjugate symmetric complex matrix defined as

$$R(k,l) = R_m(k,l)\cos\theta + iR_m(k,l)\sin\theta \quad (8)$$

where $R_m(k,l)$ is the modulus of $R(k,l)$ satisfying a 2-D normal distribution, k and l are integers within $[1, M/2-1]$, and θ is an arbitrary angle. The values in Eq. (8) meet the requirements for a conjugate symmetric complex matrix. In Eq. (7) α is the singularity factor and the FD of the image equals $3-\alpha$. So, by changing the value of α, an fBm image with a desired FD can be simulated. The inverse FFT transform can be used to quickly generate an fBm according to Eqs. (7) and (8). A simulative fBm image with $\alpha=0.9$ is shown in Fig. 4.

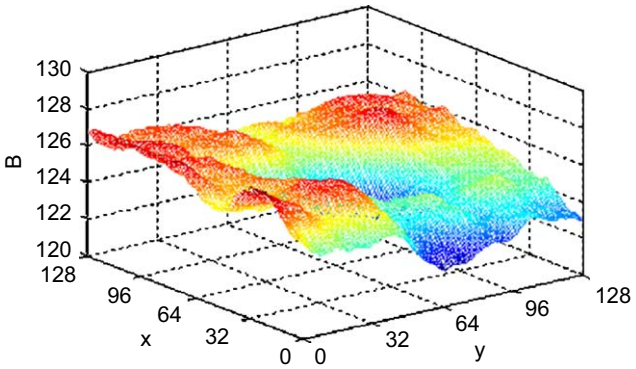

Fig. 4. A simulated fractional Brownian motion image as $\alpha=0.9$.

In the experiments, the 2-D normal distributions were used 300 times to construct 300 fBm images with the same theoretical FD. For nine FDs from 2.1 to 2.9, a total of 2700 fBm images were constructed. In addition, 300 images with $\alpha=10^{-6}$ were also used in the experiment. Their theoretical FDs were approximately equal to 2.0.

Fig. 5 shows the computational results of FD estimates of these 3000 simulative images, where the minimum, mean, and maximum values of FD estimates are computed by the improved DBC method with four different values of the parameter a in Eq. (5). The three curves, obtained by the improved DBC method with a equal to 3, 5, and 128, have better increasing characteristics than the curve obtained with a equal to 1. The curve when $a=256$ is not presented here because it is almost the same as that when $a=128$. This indicates that the computational results of FD estimates do not change obviously when a exceeds a certain value.

The FD variation range computed by the improved DBC method for different values of a are shown in Fig. 6. The FD variation range is defined as the range of the FD mean values when the theoretical FDs of the simulative fBm images change from 2.0 to 2.9. The largest FD variation range is equal to 0.554 when $a=3$. It becomes smaller than 0.5 when k exceeds 16. From Figs. 5 and 6, we can see that $a=3$ is an appropriate choice for the improved DBC method.

FD estimates of the above 3000 simulative images were computed using the improved DBC method, the original DBC method, the Pentland's method, and the Keller's method. Pentland's method

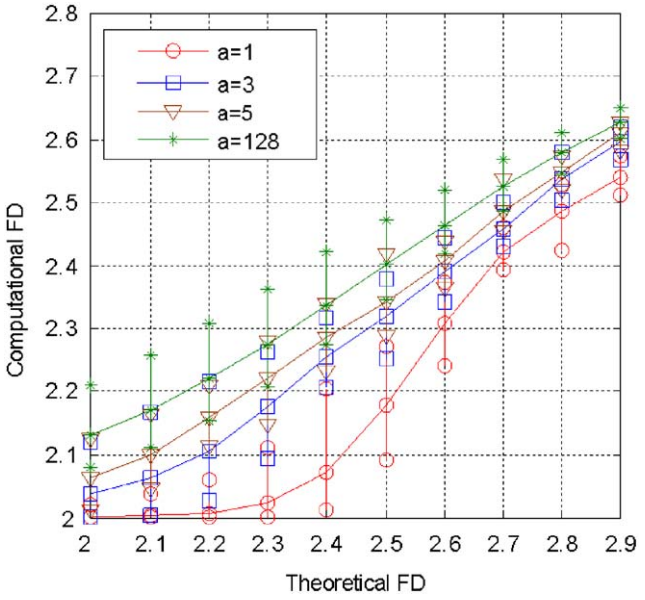

Fig. 5. FD estimates of fBm images computed by the improved DBC method with different values of *a*.

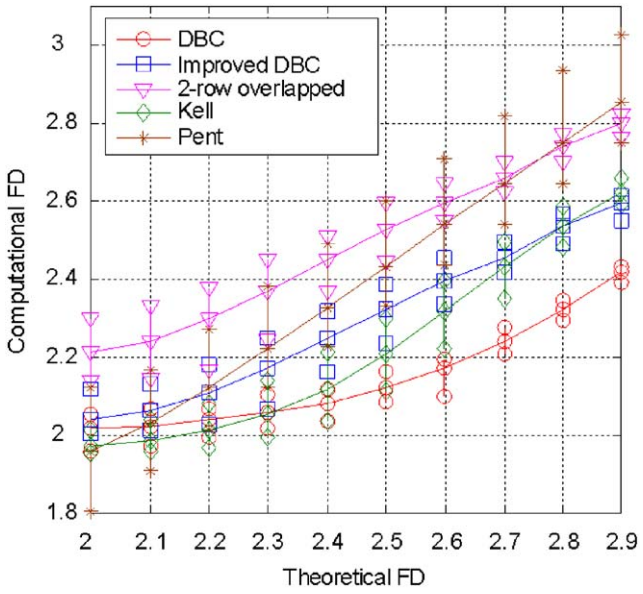

Fig. 7. Fractal dimension estimates of fBm images when their theoretical FDs increase from 2.0 to 2.9.

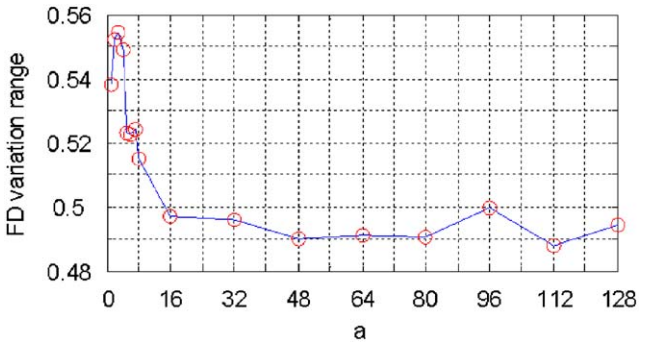

Fig. 6. FD variation range of fBm images computed by the improved DBC method with different values of *a*.

is a method of estimating FD by using the Fourier power spectrum of image intensity surface but not by counting the number of boxes that cover up total image intensity surface [1]; Keller's method introduces probability to count box number covering up total image intensity surface, which differs completely from the original DBC and the improved DBC [19]. They were selected for performance comparison due to their relatively robust performance based on our experience. The Pentland's method and the Keller's method are called briefly as Pent and Kell hereafter, respectively. As shown in Fig. 7, the improved DBC method generates FD means changing from 2.04 to 2.59, which are more accurate than that from 2.01 to 2.41 by the original DBC method. Pent generates FD means changing almost linearly, and Kell has wider variation range of FD means than the improved DBC method. However, unreasonable results of computational FDs smaller than 2 may be obtained by Pent and Kell, when simulative fBm images have small theoretical FDs. Fig. 7 also shows the improved DBC method, with two rows overlapped between two adjacent blocks, generates computational FD equal to 2.26, which is much greater than 2.0 when the estimated fBm images has theoretical FD of 2.0. Moreover, an image with all pixels having zero gray levels was used to identify if it is proper to select two rows overlapped in the modification C of the improved DBC. The theoretical FD of the image is equal to 2. The computational FD is 2.00 computed by both the DBC and improved DBC. But the improved DBC generates computational FD of 2.14 when two rows are overlapped in modification C. More computational results for three and four overlapped rows are 2.42 and 2.87, which are obviously unreasonable.

The fit error E is used to measure the least squares linear fit of the $\log(N_r)$ versus $\log(r)$. The lower fit error result from the better fit. The fit error E of points (x, y) from their fitted straight line satisfying $y = cx+d$ is defined as

$$E = \frac{1}{n}\sqrt{\sum_{i=1}^{n}\frac{cx_i + d - y_i}{1+c^2}} \tag{9}$$

where y and x denote $\log(N_r)$ and $\log(r)$ for the above approaches but not for Pent. They denote $\log(S)$ and $\log(f)$ for Pent, where $S(f_1, f_2)$ is the power spectra density distribution of a fBm image $B(x, y)$ and $f = \sqrt{(f_1{}^2+f_2{}^2)}$. Ref. [4] indicated that DBC method produced the small fit errors in computing FDs of texture images. Fig. 8 presents the minimum, mean, and maximum values of the fit errors for the simulative fBm images. First of all, Fig. 8 shows that selection of two overlapped rows is not appropriate for the improved DBC method, because the fit errors obtained by this way are obviously greater than the improved DBC method with one-overlapped row. Second, the improved DBC method has the smaller fit errors than DBC and Pent. This proves once more that the improved DBC method is more effective than these two methods. Finally, Kell has a little bit smaller fit errors than the improved DBC. But its computation complexity is much greater than DBC, presented in [12], and therefore than the improved DBC from Table 1.

It is noteworthy that the computational results of DBC, Kell, and Pent include some FD estimates that are smaller than 2.0 for the fBm images when the theoretical FD equal to 2.0, 2.1, and 2.2. The improved DBC method always generated reasonable FD estimates, which should be above 2.0 for any images in any cases. Also, considering the above analysis results of variation range of computational FDs, fit error, and computation complexity, we obtain a conclusion that the improved DBC method can provide more reasonable and accurate FD estimates for these fBm images than DBC, Kell, and Pent; meanwhile, it keeps the computational complexity as low as in DBC. It is also worth noting that the average values of estimated FDs by the improved DBC, DBC, Kell, and Pent, deviate from the theoretical

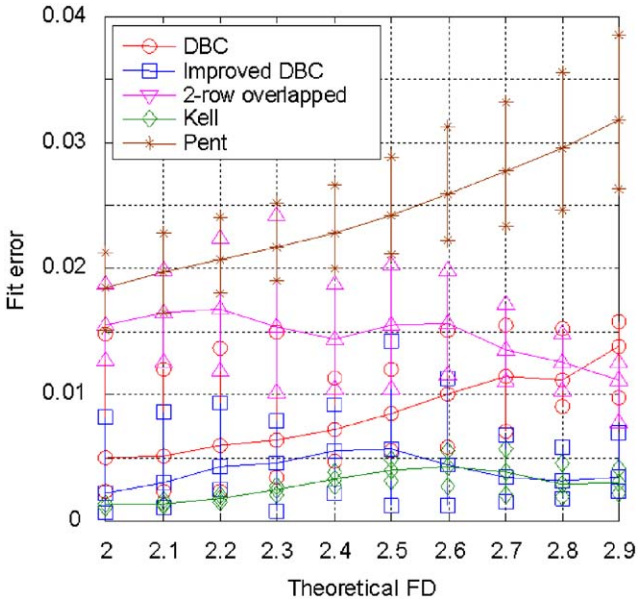

Fig. 8. Fit errors of $\log(N_r)$ vs. $\log(r)$ for fBm images.

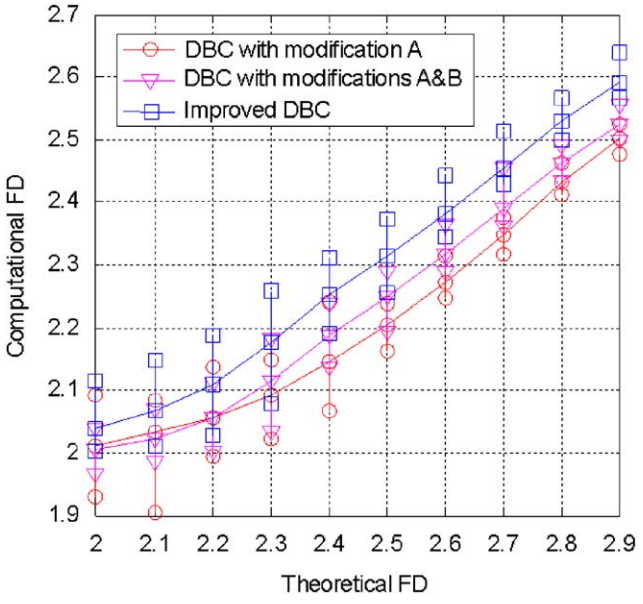

Fig. 9. Fractal dimensions of fBm images from the improved method with three modifications added in turn.

FDs. This may be due to two facts. The first is that the simulative fBm images are not the strictly ideal fractal objects and their realistic FDs differ from the expected theoretical FDs. The second is related with fit error. Fig. 8 shows that the fit error can influence the computational FDs of Pent deviating from the theoretical FDs.

To evaluate the individual contributions from the three modifications in the improved method, the box height selection only (modification A) is used to modify the DBC method firstly. The computational FDs by the first-step modified method are shown in Fig. 9. The box number calculation (modification B) and image intensity surface partition are added to modify the DBC method in turn. The computational results by the improved method and the two partially modified methods are all shown in Fig. 9. The mean values of FD estimates using only modification A and using both modifications A and

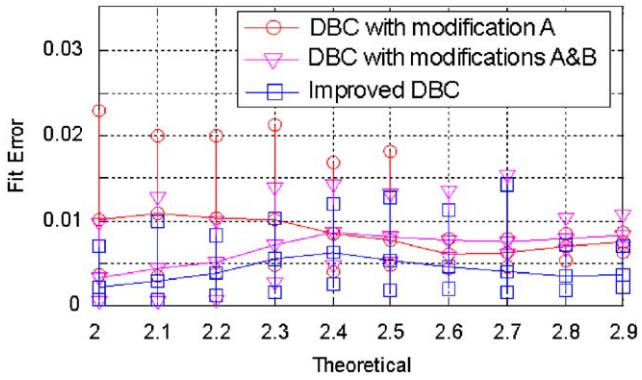

Fig. 10. Fit errors of $\log(N_r)$ vs. $\log(r)$ for fBm images corresponding to the computational dimensions in Fig. 9.

Table 2
Computational fractal dimensions of Brodatz texture images.

Texture images	DBC		Improved DBC		Pent		Kell	
	FD	Fit error	FD	Fit error	FD	Fit error	FD	Fit error
D03	2.61	0.019	2.86	0.007	2.56	0.046	2.48	0.012
D04	2.67	0.014	2.88	0.004	2.54	0.040	2.59	0.014
D05	2.40	0.024	2.65	0.010	2.02	0.030	2.30	0.010
D09	2.57	0.016	2.77	0.006	2.25	0.039	2.54	0.012
D24	2.52	0.016	2.77	0.005	2.40	0.039	2.59	0.013
D28	2.53	0.021	2.74	0.009	1.97	0.033	2.39	0.008
D33	2.30	0.014	2.49	0.007	2.51	0.039	2.22	0.001
D54	2.46	0.019	2.74	0.007	2.13	0.032	2.39	0.008
D55	2.49	0.021	2.73	0.008	2.41	0.048	2.47	0.007
D68	2.53	0.016	2.75	0.005	2.44	0.039	2.58	0.008
D84	2.58	0.020	2.82	0.005	2.13	0.039	2.54	0.014
D92	2.37	0.017	2.60	0.005	2.30	0.038	2.48	0.005

B varied from 2.01 to 2.50 and from 2.01 to 2.52, respectively, which are greater than the original DBC method, but could still be unreasonably smaller than 2.0 for the fBm images with the theoretical FD at 2.0, 2.1, and 2.2. With all three modifications, the improved DBC method yields the reasonable estimates (above 2.0) with a greater variation range. The mean fit errors, shown in Fig. 10, are smaller than the two partially modified cases. This experiment demonstrates that each modification contributes to a better performance and all three modifications together result in the ultimate advantages of the improved DBC method.

4.2. Experiment on texture images

In these experiments, 12 texture images as shown in Fig. 11 were downloaded from http://sampl.ece.ohio-state.edu/database.htm. Their FD estimates were listed in Table 2. We can see that the improved DBC method produced much smaller fit errors than DBC, Pent, and Kell; FD estimates from the improved DBC method are in the range from 2.49 to 2.88, a little bit wider than that of DBC and Kell; Pent generates one unreasonable result below 2.0 for the image D28.

To further compare these four methods, images in Fig. 11 were transformed using a linear transformation defined as

$$I^* = round\left(\frac{I - I_{\min}}{I_{\max} - I_{\min}} \times k\right) + 126 \tag{10}$$

where round $(\cdot)$ denotes the rounding operation. Here, I^* is the transformed image of an original image I, and the coefficient k varies from 1 to 50 to increase intensity variation. For a smaller value of k, more details of the image are lost and its FD decreases. When k is

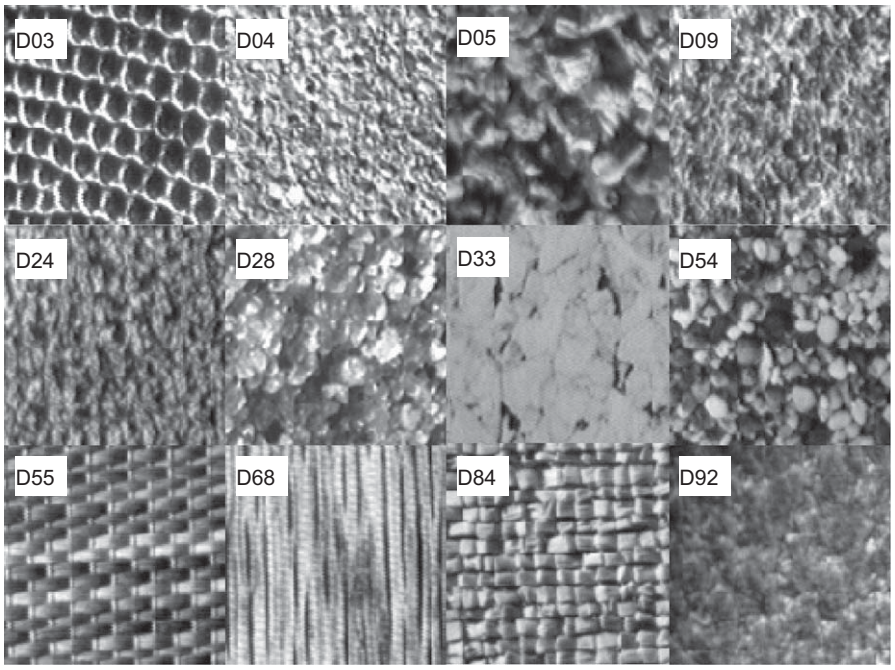

Fig. 11. Brodatz texture images.

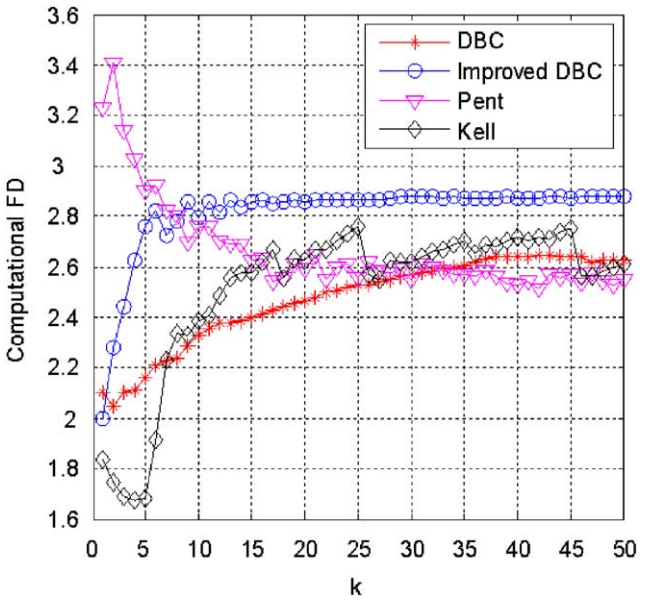

Fig. 12. Fractal dimensions of D04* transformed with different k computed by the four methods.

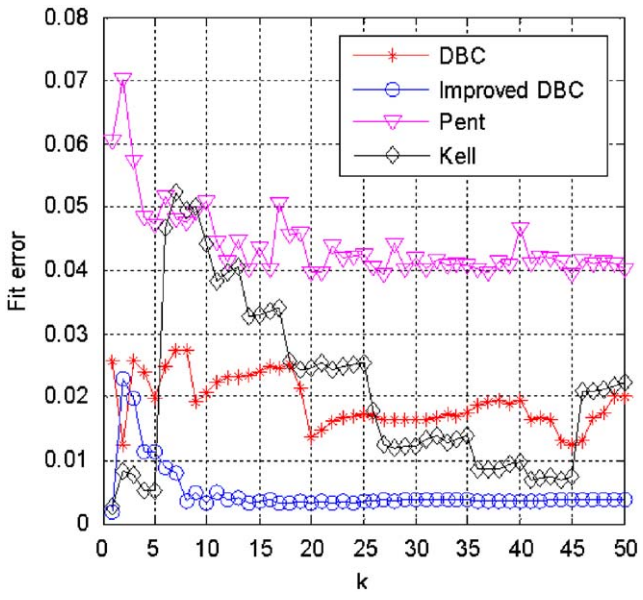

Fig. 13. Fit errors computed by the four methods for D04* transformed with different k.

large, the transformed images look similar to their original images, so their FDs should be close. Fig. 12 shows the results for D04*. We can see that the improved DBC provided greater FDs than DBC, and they were nearly equal to 2.88 after k exceeded 11; the DBC generated FDs greater than 2.6 after k was around 35; Pent demonstrated an unreasonable FD trend when k was increased to make the image rougher; Kell also yielded unreasonable and unstable results when k was increased. The smaller fit errors shown in Fig. 13 also indicate that the FD estimates from the improved DBC has higher accuracy than others.

Fig. 14 shows the transformed images in Fig. 10 when $k = 6$. In this case, they should have similar FDs as the originals. As listed in Table 3, their estimated FDs from the improved DBC were close to those of the original images in Table 1; the estimated FDs from the DBC method significantly deviated from the FDs of the original images; Pent generated completely different FDs; Kell yielded unreasonable FDs smaller than 2.0. Fig. 15 presents the results of Δ_{FD} for the DBC and improved DBC, which is defined as the difference between FDs of a transformed and the original texture images. We can see that Δ_{FD} from the improved DBC is much smaller than that from the DBC for each image. This experiment further indicates that the improved DBC method experiences much less influence from image linear transformation and can provide more robust estimates than other methods.

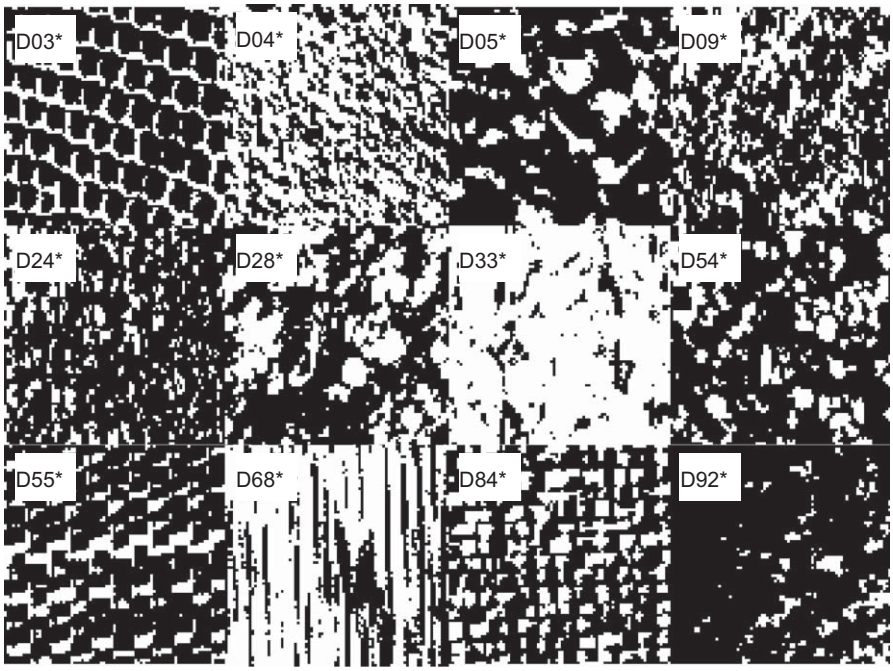

Fig. 14. Transformed smooth texture images.

Table 3
Computational fractal dimensions of the transformed texture images when $k = 6$.

Texture images	DBC		Improved DBC		Pent		Kell	
	FD	Fit error	FD	Fit error	FD	Fit error	FD	Fit error
D03*	2.19	0.023	2.65	0.008	2.88	0.054	1.79	0.052
D04*	2.21	0.025	2.82	0.009	2.92	0.052	1.92	0.047
D05*	2.12	0.016	2.56	0.014	2.61	0.040	1.62	0.056
D09*	2.18	0.023	2.66	0.011	2.80	0.045	1.79	0.055
D24*	2.17	0.023	2.61	0.008	2.87	0.049	1.80	0.060
D28*	2.15	0.019	2.65	0.012	2.50	0.039	1.71	0.052
D33*	2.04	0.008	2.27	0.011	2.96	0.048	1.69	0.046
D54*	2.15	0.019	2.61	0.013	2.57	0.039	1.72	0.054
D55*	2.16	0.020	2.64	0.009	2.95	0.059	1.69	0.060
D68*	2.16	0.019	2.63	0.011	3.01	0.050	1.85	0.051
D84*	2.18	0.023	2.69	0.011	2.75	0.044	1.81	0.054
D92*	2.12	0.015	2.35	0.010	2.91	0.054	1.73	0.059

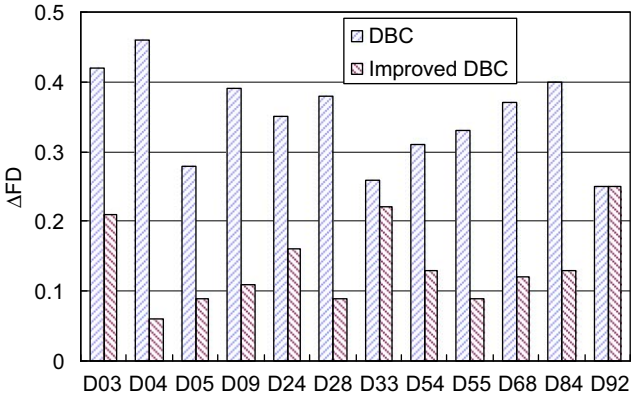

Fig. 15. Differences Δ_{FD} between the FDs of the transformed and original texture images.

4.3. Experiment on remote sensing images

Three panchromatic images of size 512×512 taken by the space-borne QuickBird multispectral sensor are shown in Fig. 16, which are about vegetation (RS1), residential (RS2), and soil (RS3) areas. The FDs of these three image scenes were computed by the four methods. As listed in Table 4, the improved DBC and Kell generated the FDs with variation range of 0.20, which was greater than 0.17 and 0.09 obtained by DBC and Pent, respectively. DBC could not separate RS2 and RS3, even though the two images were significantly different. The discrepancy in FDs between RS2 and RS3 generated by Pent and Kell was only 0.02, which was smaller than 0.05 generated by the improved DBC. This experiment demonstrates that the improved DBC method can provide better performance in image categorization, which is important for remote sensing data archiving and distribution.

The above three remote sensing images are in the same size of 512×512 pixels. Table 5 shows the computational FDs of the three images in the other two sizes of 256×256 and 128×128 pixels. The computation results of Pent were influenced by the image size most easily among the four methods; when the image sizes of the three remote sensing images were 128×128, the computational FDs of Pent became unreasonable. The computational FD differences obtained by the Kell were as large as 0.14, 0.08, and 0.13 for RS1, RS2, and RS3, respectively, which were significantly greater than the results obtained by both the original DBC and the improved DBC. In comparison of the computational results obtained by the two DBC methods, the FD differences of the improved DBC were smaller. It can be concluded that the improved DBC is less influenced by image size. This is helpful for the improved DBC used for actual image categorization.

4.4. Experiment on video degraded by atmospheric turbulence

A video frame of size 160×80 is shown in Fig. 17(a), which was degraded by atmospheric turbulence with blurred edges. Image restoration can be achieved by applying the technique proposed in Ref. [22] based on independent component analysis. Fig. 17(b) shows the restored image using five degraded frames, and Fig. 17(c)

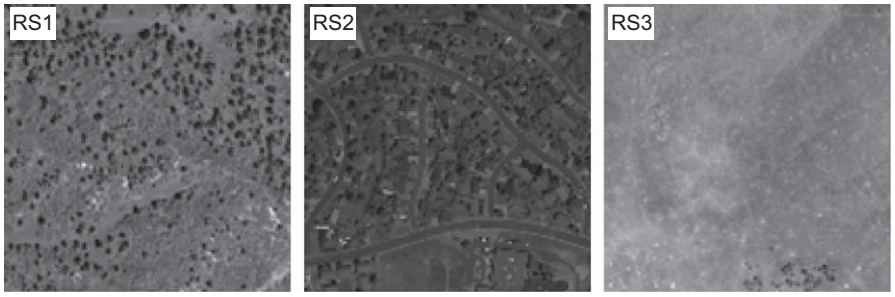

Fig. 16. Remote sensing images.

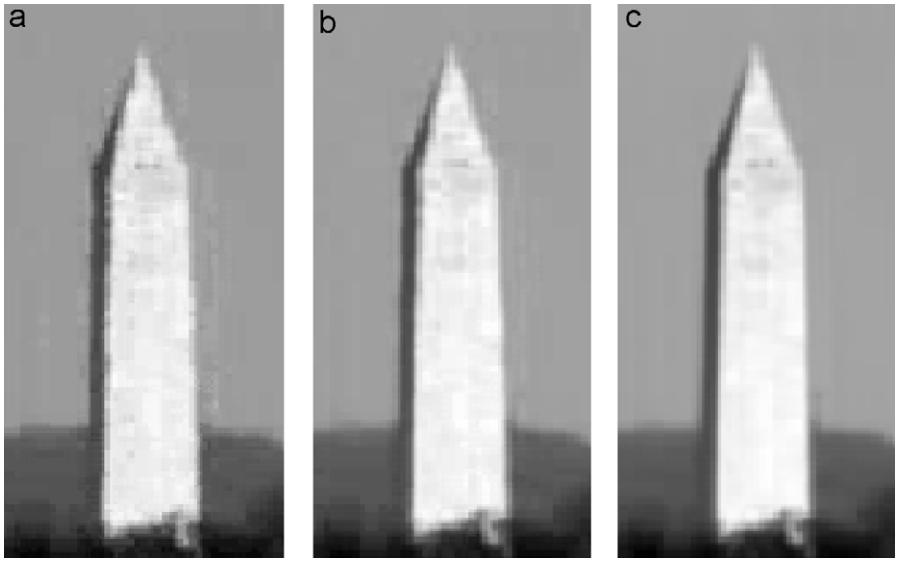

Fig. 17. Video restoration: (a) an original degraded frame, (b) restored image I, and (c) restored image II.

Table 4
Computational fractal dimensions of the remote sensing images.

Image	DBC	Improved DBC	Pent	Kell
RS1	2.51	2.62	2.25	2.43
RS2	2.34	2.42	2.32	2.23
RS3	2.34	2.47	2.34	2.25

Table 5
Computational fractal dimensions of the remote sensing images in different sizes.

Image	Image size (pixel)	DBC	Improved DBC	Pent	Kell
RS1	512×512	2.51	2.62	2.25	2.43
	256×256	2.55	2.64	2.83	2.54
	128×128	2.54	2.65	3.11	2.57
RS2	512×512	2.34	2.42	2.32	2.23
	256×256	2.36	2.43	2.98	2.28
	128×128	2.38	2.44	3.38	2.31
RS3	512×512	2.34	2.47	2.34	2.25
	256×256	2.35	2.49	2.82	2.38
	128×128	2.31	2.48	3.14	2.32

is the one using 20 degraded frames. To quantitatively evaluate the image quality, the FDs of these three images were computed by the four methods. Intuitively, an image with less turbulence should have a smaller value of FD. As tabulated in Table 6, the Pent and Kell generated unreasonable results smaller than 2.0 for the image in Fig. 17(c), which may be due to the fact that the monument is an artificial object and the restored image II is in very good quality with smooth intensity surface (as similar as presented in Fig. 7 where Kell and Pent generated unreasonable results of FD for smooth intensity surfaces). The improved DBC provided reasonable values of FD to manifest the three images with different quality, while the original DBC could not identify the difference between the original frame and the restored image I.

Table 6
Computational fractal dimensions of the degraded and restored images.

Image	DBC	Improved DBC	Pent	Kell
Original	2.06	2.17	2.46	2.06
Restored I	2.06	2.15	2.24	2.00
Restored II	2.05	2.14	1.97	1.98

5. Conclusions

A modified box-counting-based method is proposed to estimate the FD of an image. To improve the estimate accuracy, it is required to use the smallest number of boxes to completely cover the image intensity surface at each specific box dimension. Three major contributions are included in our work to meet this requirement: the first one is about the box height selection that provides a finer measure

for counting the box numbers; the second lies in the determination of the number of boxes that guarantees the least number of boxes can be obtained to cover every block for each specific scale; and the last one is about completely covering the image intensity surface using overlapping blocks without violating the basic requirements of box-counting dimension estimation. Experimental results demonstrate that they can bring about better performance. Compared to the original DBC method, Pentland's method, and Keller's method, the improved DBC method can provide FD estimates with smaller fit errors.

FD estimation is employed for practical applications: the experiment on texture image classification shows that the improved DBC is much less influenced by linear image transformation than the other three methods, thereby providing more robust estimates; the experiment on remote sensing images demonstrates that the improved DBC method is more efficient in characterizing natural images and can be more useful in image categorization; the experiment on video restoration shows that the FD from the improved DBC method can be used for no-reference image quality assessment.

Acknowledgments

The authors acknowledge the funding of the 111 Project from the Ministry of Education, China (B08036). The Program for New Century Excellent Talents in University (NCET-06-0763) is also appreciated for supporting this work.

References

[1] A.P. Pentland, Fractal-based description of natural scenes, IEEE Transactions on Pattern Analysis and Machine Intelligence 6 (1984) 661–674.
[2] B.B. Mandelbrot, The Fractal Geometry of Nature, Freeman, San Francisco, CA, 1982.
[3] H.O. Peitgen, H. Jurgens, D. Saupe, Chaos and Fractals: New Frontiers of Science, first ed, Springer, Berlin, 1992.
[4] B.B. Chaudhuri, N Sarker, Texture segmentation using fractal dimension, IEEE Transactions on Pattern Analysis and Machine Intelligence 17 (1995) 72–77.
[5] S. Liu, S. Chang, Dimension estimation of discrete-time fractional Brownian motion with applications to image texture classification, IEEE Transactions on Image Processing 6 (1997) 1176–1184.
[6] T. Ida, Y. Sambonsugi, Image segmentation and contour detection using fractal coding, IEEE Transactions on Circuits System Video Technology 8 (1998) 968–977.
[7] G. Neil, K.M. Curtis, Shape recognition using fractal dimension, Pattern Recognition 30 (12) (1997) 1957–1969.
[8] K.H. Lin, K.M. Lam, W.C. Siu, Locating the eye in human face images using fractal dimensions, IEE Proceedings on Vision, Image and Signal Processing 148 (2001) 413–421.
[9] P. Asvestas, G.K. Matsopoulos, K.S. Nikita, A power differentiation method of fractal dimension estimation for 2-D signals, Journal of Visual Communication and Image Representation 9 (1998) 392–400.
[10] A.S. Balghonaim, J.M. Keller, A maximum likelihood estimate for two-variable fractal surface, IEEE Transactions on Image Processing 7 (1998) 1746–1753.
[11] L. Gangepain, C. Roques-Cames, Fractal approach to two dimensional and three dimensional surface roughness, Wear 109 (1986) 119–126.
[12] N. Sarker, B.B. Chaudhuri, An efficient differential box-counting approach to compute fractal dimension of image, IEEE Transactions on Systems, Man, and Cybernetics 24 (1994) 115–120.
[13] S. Buczkowski, S. Kyriacos, F. Nekka, L. Cartilier, The modified box-counting method: analysis of some characteristics parameters, Pattern Recognition 3 (1998) 411–418.
[14] W.-S. Chen, S.-Y. Yuan, C.-M. Heieh, Two algorithms to estimate fractal dimension of gray-level images, Optical Engineering 42 (2003) 2452–2464.
[15] G. Du, T.S. Yeo, Novel multifractal estimation method and its application to remote image segmentation, IEEE Transactions on Geoscience and Remote Sensing 40 (2002) 980–982.
[16] S. Novianto, Y. Suzuki, J. Maeda, Near optimum estimation of local fractal dimension for image segmentation, Pattern Recognition Letters 24 (2003) 365–374.
[17] S. Xu, Y. Weng, A new approach to estimate fractal dimensions of corrosion images, Pattern Recognition Letters 27 (2006) 1942–1947.
[18] S. Peleg, J. Naor, R. Hartley, D. Avnir, Multiple resolution texture analysis and classification, IEEE Transactions on Pattern Analysis and Machine Intelligence 6 (1984) 518–523.
[19] J. Keller, R. Crownover, S. Chen, Texture description and segmentation through fractal geometry, Computer Vision, Graphics, and Image Processing 45 (1989) 150–166.
[20] W. Xie, W. Xie, Fractal-based analysis of time series data and features extraction, Chinese Signal Processing Journal 13 (1997) 98–104.
[21] L. Yu, D. Zhang, K. Wang, W. Yang, Coarse iris classification using box-counting to estimate fractal dimensions, Pattern Recognition 38 (2005) 1791–1798.
[22] I. Kopriva, Q. Du, H. Szu, Independent component analysis approach to image sharpening in the presence of atmospheric turbulence, Optics Communications 233 (2004) 7–14.

About the Author—JIAN LI received the M.S. and Ph.D. degree in electrical engineering in 1997 and 2001, from Chongqing University, Chongqing, China. Currently, he is an Associate Professor and the Head of High Voltage and Insulation Technology Department at Chongqing University. He is now a Visiting Professor at the High Voltage Laboratory of Mississippi State University in US. His major research interests include online detection of insulation condition of electrical devices, partial discharges, and insulation fault diagnosis for high voltage equipment. He is an author and coauthor of more than 20 journal papers and 25 papers published in proceedings of international conferences. Dr. Li is a member of IEEE Dielectric and Electrical Insulation Society, IEEE Power Society and Chinese Society of Electrical Engineering.

About the Author—QIAN DU received her Ph.D. degree in electrical engineering from University of Maryland Baltimore County in 2000. She is currently an Associate Professor in the Department of Electrical and Computer Engineering, Mississippi State University. Her research interests include digital image processing, pattern recognition, data compression, neural networks, and remote sensing. She is a member of IEEE, SPIE, ASPRS, and ASEE.

About the Author—CAIXIN SUN received the B.S. degree in electrical engineering in 1969 from Chongqing University. He is professor of Chongqing University and was Vice President of Chongqing University. Currently, he is a member of the Chinese Engineering Academy and the Director of Electrical Power Engineering Committee of National Science Foundation of China. His research interests include online detection of insulation condition and insulation fault diagnosis for high voltage equipment, discharge mechanism of outdoor insulation in complicated environment, and high voltage technique applied in biomedicine. He is an author and coauthor of over 200 publications.