Pattern Recognition Letters 98 (2017) 53–58

Contents lists available at ScienceDirect

Pattern Recognition Letters

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

Enhancement of the Box-Counting Algorithm for fractal dimension estimation

Gun-Baek So [a], Hye-Rim So [a], Gang-Gyoo Jin [b,*]

[a] Ocean Science and Technology School, KIOST- KMOU, Busan 49112, South Korea
[b] Division of IT, Korea Maritime and Ocean University, Busan 49112, South Korea

ARTICLE INFO

Article history:
Received 19 September 2016
Available online 19 August 2017

Keywords:
BC method
Fractal dimension
Sampling method
Fractional box counting

ABSTRACT

The box-counting (BC) method is frequently used as a measure of irregularity and roughness of fractals with self-similarity property due to its simplicity and high reliability. It requires a proper choice of the number of box sizes, corresponding sizes, and size limits to guarantee the accuracy of the fractal dimension estimation. Most of the existing BC methods utilize the geometric-step method, which causes a lack of fitting data points and wasted pixels for images of large size and/or arbitrary size. This paper presents a BC algorithm in combination with a novel sampling method and fractional box-counting method which will allow us to overcome some of limitations evident in the conventional BC method. The new sampling method introduces a partial competition based on the coverage of box sizes and takes more number of box sizes than the geometric-step method. To circumvent the border problem occurring for images of arbitrary size, the fractional box-counting method allows the number of the boxes to be real, rather than integer. To show its feasibility, the proposed method is applied to a set of fractal images of exactly known fractal dimension.

1. Introduction

Fractals are complex objects or patterns and characterized by properties such as self-similarity and infinite detail. When a part of a fractal object scaled in each spatial direction is statistically similar to itself, it is called *self-similarity* [1]. The never-ending repetition of a simple structure is called *recursiveness*. Fractal dimension is an effective tool for describing the inherent irregularity of natural objects. Pentland [2] demonstrated that fractal dimension is highly correlated with human perception of fractal surface; the rougher the surface appears, the larger is the fractal dimension.

Since the establishment of fractal theory, various methods of estimating the fractal dimension such as the box-counting (BC) method, triangular prism method, and fractional Brownian motion method have been suggested [3]. Among these, the BC method is the most frequently used in various application fields due to its relative simplicity and reliability [4-7]. The BC method has been applied to recognize human iris image [4], analyze texture images [5], measure the complexity of the cerebral cortex surface of a fetus [6], estimate the river water flow rate in a basin [7], and calculate the complexity of a distributed network and to create a virtual distributed network by using an extended fractal model [8].

Recently there have been several works to increase the precision of the BC method. Foroutan-pour et al. [9] discussed the preparation of an image for box-counting analysis and the determination of the appropriate size of a rectangular frame surrounding a fractal. Milošević and Elston [10] suggested a method of eliminating regression analysis data having nonlinear properties by adequately controlling the box size and applied the method to estimate the fractal dimension of 2D images from caudate nucleus. Kaewaramsri and Woraratpanya [11] suggested the triangular box-counting method where the conventional square box is divided into two triangles to increase the precision of box-counting.

Most of the existing BC methods suggested so far utilize the geometric-step (GS) method [12] where the step size is limited to a power of 2 so that the pixels of the handled image may not be wasted. Unfortunately, application of that method to images of large size usually gives insufficient points for data fitting. When the BC method is applied to images of size $M \times N$ pixels (M and N may not powers of 2), decrease of the performance is unavoidable due to some possible pixel waste in the calculation.

In this paper, we present a novel BC method, which improves the accuracy of fractal dimension estimation by solving some problems of the conventional BC method and is applicable to images with arbitrary sizes by allowing fractional box counting for wasted pixels. A new sampling method which employs a partial competition based on the coverage of box sizes and takes more number of

* Corresponding author.
E-mail address: ggjin@kmou.ac.kr (G.-G. Jin).

http://dx.doi.org/10.1016/j.patrec.2017.08.022
0167-8655/

box sizes than the GS method is introduced. To circumvent the border problem occurring for images of arbitrary size, the fractional box-counting method allows the number of the boxes to be real, rather than integer. The performance of the proposed BC method is verified with a set of 120 deterministic fractal images of known theoretical dimension and compared that of the conventional BC method.

2. Conventional BC method

Mandelbrot [1] defined that a set in an Euclidean space is said to be *self-similar* if it is the union of $N(r)$ distinct subsets, each of which is a copy of the original scaled down by a ratio 1/r in each spatial direction. The fractal dimension of an object is defined by

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

where $N(r)$ is the least number of boxes of length r needed to completely cover the object.

Eq. (1) can be directly applicable to deterministic fractals such as the Koch snowflake or the Sierpinski triangle. However, for natural (non-deterministic) fractals, an appropriate method is needed to count $N(r)$. Among a number of approaches, the BC algorithm, introduced by Gangepain and Roques-Carmes [13], is found to be the most desirable method of approximate fractal dimension estimation. Given a binary (black & white) image of $M \times M$ pixels, where M is a power of 2, Eq. (1) can be rewritten from the relation between box size (step size) δ and M, that is, $\delta = rM (0 < r < 1)$:

$$D \approx \frac{\log(N(\delta))}{\log(1/\delta)}. \tag{2}$$

Hence, the overall procedure of the BC method can be summarized into three steps: First, a set of box sizes δ for laying grids on the image by using a sampling method is generated. Each grid becomes a box of size $\delta \times \delta$. Then, for each δ, the number of boxes $N(\delta)$ needed to cover the object completely is counted. Finally, D is obtained from the slope of points $(\log(1/\delta), \log(N(\delta)))$.

3. Proposed BC method

3.1. Novel sampling method

One of key issues that many fractal estimation algorithms including the BC method have faced is the choice of δ, the number of δ, and δ range, that is, a sampling strategy. The estimation accuracy is sensitive to the sampling strategy, the improper use of which leads to problems while estimating D [14,15]. There are three non-overlapping regular grid methods; geometric-step method, arithmetic-step method and divisor-step method [12,16]. One of the most widely used methods is the GS method, adopted by most of fractal estimation algorithms. The GS method uses a set of powers of 2 defined by $\Delta_{GS} = \{1,2,\ldots,M/2\}$. The right extreme value M is not of great use ($N(\delta)=1$ when $\delta = M$).

Unfortunately, application of that method to a large size of M usually does not give enough points for the least square linear fit. In general, the size of images acquired in real-world situation, such as coastlines, rivers, chains of mountains, and parts of living organisms is actually arbitrary. Application of that method to an image of $M \times N$ may waste pixels in the calculation and degrade the estimation accuracy. A novel sampling method is herein proposed to resolve some of these problems on the basis of the intuitive observation that as the better steps and the more regression data are selected, the higher is estimation accuracy.

Given an image of size $M \times N$ pixels for general cases, where M and N are may not be powers of 2, let us define a coverage ratio $CR(\delta)$, which is the ratio of pixel utilization for an image at each step δ as:

$$CR(\delta) = \frac{\mathrm{int}(\frac{M}{\delta})\delta \times \mathrm{int}(\frac{N}{\delta})\delta}{M \times N} (\delta \in \Delta), \tag{3}$$

where $\mathrm{int}(\frac{M}{\delta})\delta \times \mathrm{int}(\frac{N}{\delta})\delta$ denotes the number of pixels covered at δ, and Δ the collection of δ selected by a sampling method. In addition, with respect to all box sizes, the average coverage ratio $\overline{CR}$ is defined as:

$$\overline{CR} = \frac{1}{|\Delta|} \sum_{\delta \in \Delta} CR(\delta). \tag{4}$$

In the case of an image of size $M \times N$, its sampling should be performed with respect to the length of the short side. Let us $L = \min(M,N)$ and $\ell = \mathrm{int}(\log_2(L))$, where int() is an operator taking the integer part. The proposed sampling method, called the modified GS (MGS) method, is an extended version of the GS method by incorporating coverage of δ and designed to be increased the overall pixel utilization and at the same time be ensured more data than the GS method. It setups the selection pool of δ as:

$$\Delta_{POOL} = \left\{ \delta | \delta = round\left(2^{(k+\frac{1}{2})}\right) \leq \frac{L}{2}, \quad \mathrm{k} = 1, 2, \ldots, \ell - 1 \right\}, \tag{5}$$

where round() is a rounding-off operator. Then, it selects the winners of n_{SEL} as in Eq. (6) to constitute the resulting set Δ_{MGS} in conjunction with Δ_{GS}.

$$n_{SEL} = \min[n_{POOL}, round(\eta . n_{GS})], \tag{6}$$

where $n_{GS} = |\Delta_{GS}| = \ell$, $n_{POOL} = |\Delta_{POOL}| = \ell - 1$, and $\eta (\geq 0)$ is a user-defined positive constant which directly affects on the performance.

The following algorithm is the overall description of the MGS method:

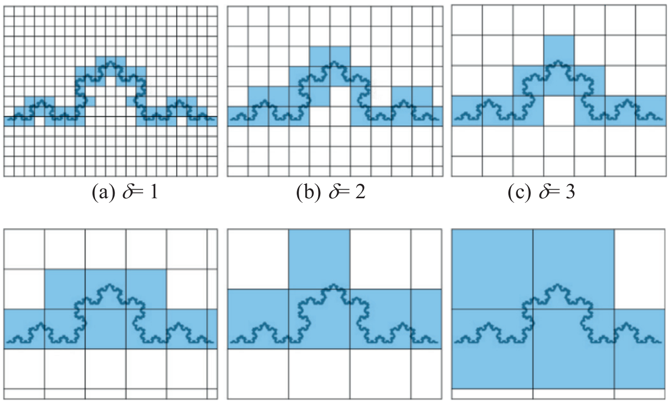
Algorithm 1 The MGS method.

1: $L = \min(M, N)$, $\ell = \mathrm{int}(\log_2(L))$
2: Setup $\Delta_{GS} = \{\delta \mid \delta = 2^{k-1}, k = 1,2,\ldots,\ell\}$;
3: Setup $\Delta_{POOL} = \{\delta \mid \delta = round(2^{(k+\frac{1}{2})}) \leq \frac{L}{2}, \mathrm{k} = 1,2,\ldots, \ell\text{-}1\}$;
4: **for** $k = 1$ to $\ell - 1$ do
5: Get $\delta_k \in \Delta_{POOL}$;
6: Calculate $CR(\delta_k)$ using Eq. (3);
7: **end**
8: Setup a matrix $\boldsymbol{A} = [CR(\delta_k), \delta_k]$ $(k = 1,2,\ldots, \ell - 1)$;
9: After sorting in descending order based on the first column, take δ_k of n_{SEL} from the second column of $\boldsymbol{A}$ to setup Δ;
10: Return $\Delta_{MGS} = \Delta_{GS} \cup \Delta$, sorted in ascending order;

For example, applying the MGS method with $\eta = 0.6$ to an image of 17×21 gives $L = \min(17,21) = 17$, $\ell = \mathrm{int}(\log_2(L)) = 4$, $\Delta_{GS} = \{1,2,4,8\}$, $\Delta_{POOL} = \{3,6\}$, $CR(\delta) = \{0.882, 0.605\}$ for $\delta \in \Delta_{POOL}$, $n_{SEL} = \min(2, round(0.6 \times 4)) = 2$ and results in $\Delta_{MGS} = \{1, 2, 3, 4, 6, 8\}$. Hence, $\overline{CR} = 0.833$.

3.2. Fractional box-counting

Since not Algorithm 1 all sampling methods can use entire pixels for a $M \times N$ image, we suggests a fractional box-counting method to resolve the wasted pixel problem. The image is divided to make each block to be a square or a rectangle as shown in Fig. 1. Integer counting is performed in blocks of a $\delta \times \delta$ size as in the conventional BC method. In other words, if the (i,j)th block contains at least one pixel of the fractal, $n_{ij} = 1$; otherwise, $n_{ij} = 0$. The other blocks are counted by one of the following equations; other-



(a) δ= 1 (b) δ= 2 (c) δ= 3

(d) δ= 4 (e) δ= 6 (f) δ= 8

Fig. 1. Sampling of the image of 17×21.

Table 1
Box sizes and box numbers.

Box size δ	1	2	3	4	6	8
Box Count $N(\delta)$	56	21.19	10	8.04	4.24	4.61

wise, $n_{ij}=0$:

$$n_{ij} = \begin{cases} \min(\frac{p_{ij}}{\bar{p}_\delta}, 1) \times \frac{(M-m\delta)}{\delta} & (M > m\delta, N = n\delta) \\ \min(\frac{p_{ij}}{\bar{p}_\delta}, 1) \times \frac{(N-n\delta)}{\delta} & (M = m\delta, N > n\delta) \\ \min(\frac{p_{ij}}{\bar{p}_\delta}, 1) \times \frac{(M-m\delta)(N-n\delta)}{\delta^2} & (M > m\delta, N > n\delta) \end{cases}, \tag{7}$$

where p_{ij} denotes the number of fractal pixels within the (i,j)th block and $\bar{p}_\delta$ the average number of fractal pixels within the blocks of size $\delta \times \delta$ which are counted as 1; $m = \text{int}(M/\delta)$ and $n = \text{int}(N/\delta)$.

Application of the fractional counting method to an image of 17×21 shown in Fig. 1 gives $N(\delta)$ in Table 1.

For example, in Fig. 1(f) where $\delta=8$, since $m = \text{int}(21/8) = 2$ and $n = \text{int}(17/8) = 2$, four blocks are counted as 1, and the (2,3)th block of 8×5 is counted as a real number. The fact that the average number of fractal pixels within four blocks is calculated as $\bar{p}_\delta = 11.25$ and $p_{ij} = 11$ gives

$$n_{2,3} = \min(\frac{p_{ij}}{\bar{p}_\delta}, 1) \times \frac{(N - n\delta)}{\delta} = \frac{11}{11.25} \times \frac{5}{8} = 0.61$$

Hence, $N(\delta) = 4 + 0.61 = 4.61$.

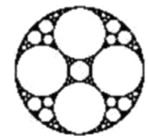
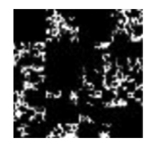
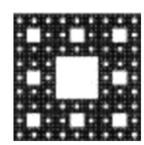
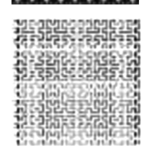
Once $N(\delta)$ is obtained for each δ, the fractal dimension is calculated from the least square linear fit of $\log(1/\delta)$ against $\log(N(\delta))$ in the proposed method.

4. Experiment and discussion

4.1. Test images

To check the validity of the proposed BC algorithm employing the MGS method and fractional counting method and compare its performance with that obtained by the conventional BC method, eight fractals of known theoretical dimensions between 1 to 2 are intensively chosen. Since the estimate of a fractal dimension may be slightly dependent on the levels of the fractals used, a series of images of size 128×128, 256×256, and 512×512 pixels for each set of fractals and for each set of levels are generated. A total of 120 images (= 8 fractals × 5 levels × 3 sizes) were drawn by using MATLAB with uniform thickness of one pixel and by minimizing the area covered by a square frame. Table 2 summarizes the images used for the experiment as well as the theoretical dimensions and levels.

Table 2
Test images for experiments.

No.	Name of figure	Figure	Dimension	Level
1	Koch snowflake		1.262	4~8
2	Apollonian Gasket		1.328	3~7
3	Vicsek fractal		1.465	3~7
4	Sierpinski triangle		1.585	5~9
5	Rand cantor		1.678	Five seeds
6	Koch curve 85°		1.785	5~9
7	Sierpinski Carpet		1.893	3~7
8	Hilbert curve		2	6~10

4.2. Determination of η

As mentioned in the previous section, one parameter η affects the performance of the proposed method. η determines how many box sizes it takes from δ's of ℓ-1. It is determined through the experiment performed with the images of 120. An evaluation function used to measure the estimation performance is the mean absolute error (MAE) between the theoretical dimension D_t and estimated dimension D_i as follows:

$$MAE = \frac{1}{w}\sum_{i=1}^{w} |D_t - D_i|, \tag{8}$$

where w denotes the number of fractals used in the experiment.

Fig. 2 shows the MAE of the proposed method, calculated with respect to $\eta = 0 \sim 1$. It can be seen in the figure that, when $0 < \eta \leq 0.6$, the performance of the proposed method is better than that of the BC method ($\eta=0$) and the best performance is obtained at $\eta = 0.6$. The performance is worse than that of the BC method when $\eta > 0.6$ since many box sizes having a low pixel utilization rate are included. It is quite interesting to note that, although the image size is $M \times M$ (M is a power of 2) and 100% coverage is secured in the conventional BC method employing the GS method,

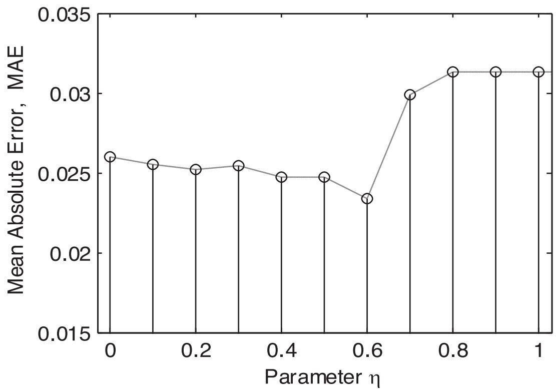


Fig. 2. *MAE* of the proposed method computed with different values of η.

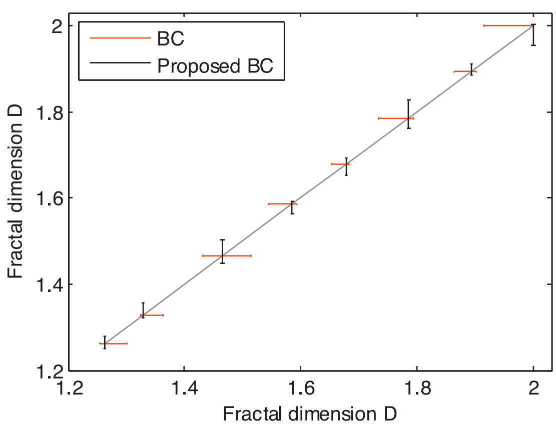


Fig. 3. Mean error bars between each theoretical fractal dimension and the estimates.

Table 3
Estimates of the images of $M \times M$ obtained by the two methods.

No	Geometry	Fractal dimension (Standard deviation)		
		Theoretical	BC	Proposed BC
1	Koch snowflake (Opened form)	1.262	1.292 (0.023)	1.267 (0.018)
2	Apollonian gasket	1.328	1.378 (0.015)	1.368 (0.015)
3	Vicsek fractal	1.465	1.464 (0.046)	1.465 (0.029)
4	Sierpinski triangle	1.585	1.553 (0.040)	1.583 (0.018)
5	Rand cantor	1.678	1.655 (0.019)	1.677 (0.022)
6	Koch curve 85°	1.785	1.736 (0.047)	1.788 (0.041)
7	Sierpinski Carpet	1.893	1.876 (0.030)	1.901 (0.016)
8	Hilbert curve	2.000	1.913 (0.092)	1.974 (0.063)

maintaining the number of box sizes 1.6 times of n_{GS} gives better performance at the slight expense of $\overline{CR}$.

4.3. Experiment with images of $m \times m$ pixels

The first experiment is performed on the fractal images of 120 to verify the performance of the proposed method. For each set of fractals, fifteen images (=5 levels $\times$3 sizes) are used to calculate their fractal dimensions, and the mean and standard deviation are also calculated. The results of the proposed method are compared with those obtained by the BC method. Table 3 depicts the results obtained by the two methods as well as theoretical dimensions.

These results confirm that the proposed method gives more accurate estimates close to the theoretical dimensions than the BC method with mostly lower standard deviation. A further evaluation for all the 120 fractals indicates that the *MAE* of 0.0234 by the proposed method is more accurate than the *MAE* of 0.0402 by the BC method. Fig. 3 also shows both horizontal (BC) and vertical (Proposed BC) error bars in the averages. The left (lower) and right (upper) lengths of the horizontal (vertical) error bars at each point mean negative and positive mean errors, respectively. The results of the proposed method yield less deviation from the theoretical fractal dimension.

4.4. Experiment with images of $m \times n$ pixels

In the previous experiments, the two methods were guaranteed 100% coverage of the entire window at any given step, but in real-world situation, sometimes one deals with the window of varying sizes. Hence, to evaluate the performance of the two methods with images of $M \times N$, we select the Sierpinski triangle (level 5 to 9) and Sierpinski carpet (level 3 to 7) of 256×256 pixels.

While changing N from 128 to 256 with a fixed M (= N, 128, 191 or 256), subsets of each original image are generated by starting from the top left corner and their estimated fractal dimensions are averaged to calculate *MAEs*. The plots of *MAE* vs. window size obtained from the five images are shown in Fig. 4 for Sierpinski triangle and in Fig. 5 for Sierpinski carpet.

It is found from the figures that the proposed method consistently gives higher performances than the BC method across all window size. However, the performances of the BC method employing the GS method are very sensitive to the image size, in other words, the coverage ratio, by making this method far less desirable for fractal dimension estimation of images of $M \times N$. The *MAEs* of the BC method fluctuate a lot across the window size and this pattern is similar to the average waste ratio, $\overline{WR}$ (=1-$\overline{CR}$). It can be seen that the worst performances occur when the window size is 191, which gives the lowest coverage.

5. Conclusion

The GS method, the most frequently adopted sampling method, incorporates the BC algorithm for fractal dimension estimation of binary images. However, the drawbacks of this method is that, when the window size is large, the number of points for data fitting are not enough and its application to images of arbitrary size causes border problem, that is, waste of pixels.

In this study, we suggested a novel BC method employing both the MGS sampling method and fractional box-counting method, which overcomes such problems associated with the conventional BC method. The MGS method constitutes a set of box sizes on the basis of pixel utilization ratio to ensure more data points than the GS method. The fractional box-counting method resolves the problem if at least one fractal pixel is contained within discarding blocks. η, which affects the performance, was determined as a value that minimizes *MAE* by using the fractals for which theoretical fractal dimensions are known.

Comparison of the proposed method with the conventional BC method on a set of 120 images with known fractal dimension showed that the accuracy was significantly improved. One interesting finding is the fact that the accuracy of the BC method was higher when the number of box sizes is maintained at a value about 1.6 times greater than that of the GS method even though the image size is a power of 2.

Applications of the modified GS method and fractional box-counting method to other fractal estimation methods including the

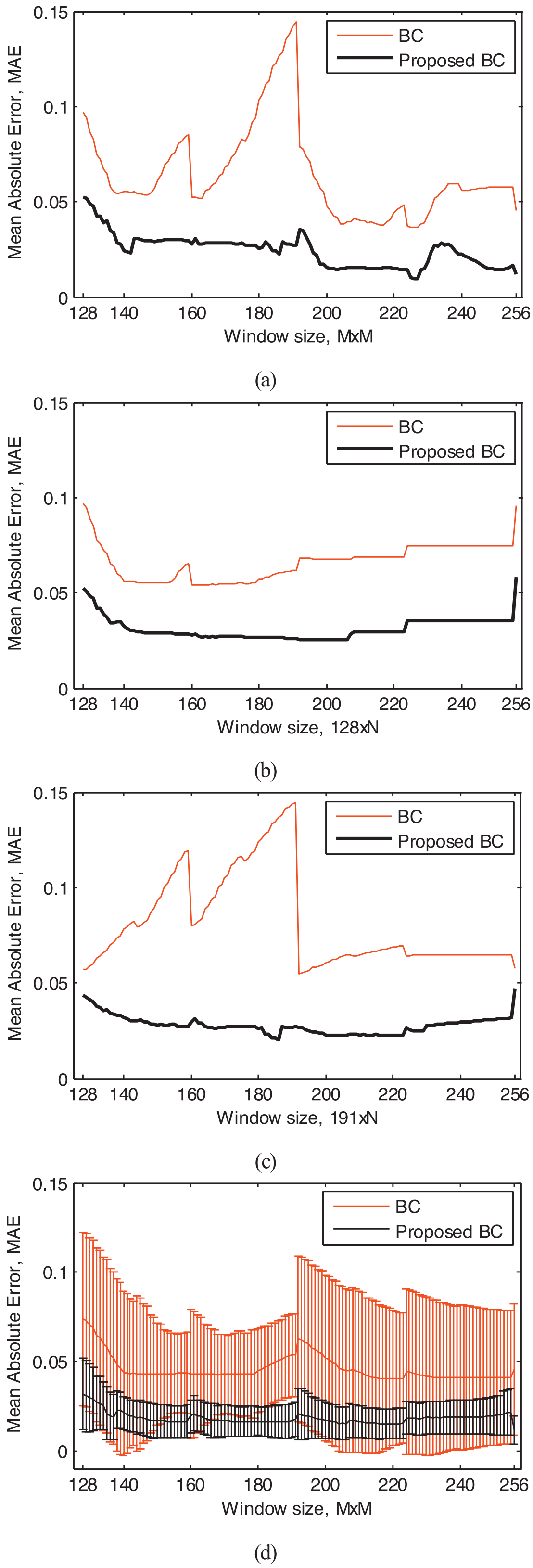


Fig. 4. *MAEs* of the two methods on Sierpinski triangle images. (d) error bar plots of *MAEs* and standard deviations.

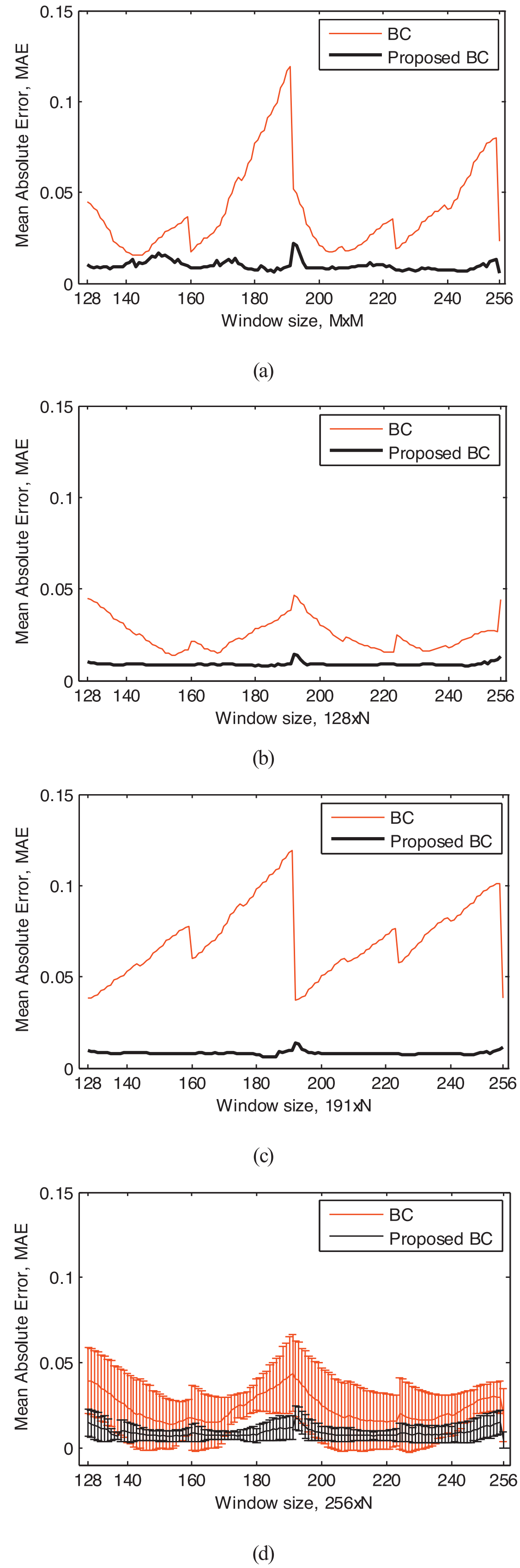


Fig. 5. *MAEs* of the two methods on Sierpinski carpet images. (d) error bar plots of *MAEs* and standard deviations.

differential box-counting method should be researched as a future works.

Acknowledgements

The authors wish to thank the reviewers and the associate editor, for their valuable comments.

References

[1] B.B Mandelbrot, How Long Is the Coast of Britain? Statistical Self-Similarity and Fractional Dimension, Science 156 (3775) (1967) 636–638.
[2] A. Pentland, Fractal-Based Description of Natural Scenes, IEEE Trans. Pattern Anal. Mach. Intell. PAMI-6 (6) (1984) 661–674.
[3] R. Lopesa, N. Betrouni, Fractal and multifractal analysis: A review, Med. Image Anal. 13 (4) (2009) 634–649.
[4] L. Yu, D. Zhang, K. Wang, W. Yang, Coarse iris classification using box-counting to estimate fractal dimensions, Pattern Recognit. 38 (11) (2005) 1791–1798.
[5] J. Li, Q. Du, C. Sun, An improved box-counting method for image fractal dimension estimation, Pattern Recognit. 42 (11) (2009) 2460–2469.
[6] K.K. Shyu, Y.T. Wu, T.R. Chen, H.Y. Chen, H.H. Hu, W.Y. Guo, Measuring Complexity of Fetal Cortical Surface From MR Images Using 3-D Modified Box-counting Method, IEEE Trans. Instrum. Meas. 60 (2) (2011) 522–531.
[7] M. Lin, L. Chen, Y. Ma, Research on stream flow series fractal dimension analysis and its relationship with soil erosion, in: IEEE Int. Symp. On Geoscience and Remote Sensing (IGARSS), Melbourne, Australia, 2013, pp. 1821–1823.
[8] F. Barakou, D. Koukoula, N. Hatziargyriou, A. Dimeas, Fractal geometry for distribution grid topologies, in: IEEE Eindhoven PowerTech Conference, 2010, pp. 1–6.
[9] K. Foroutan-pour, P. Dutilleul, D.L. Smith, Advances in the implementation of the box-counting method of fractal dimension estimation, Appl. Math. Comput. 105 (2-3) (1999) 195–210.
[10] N.T. Miloševi, G.N. Elston, Box-count analysis of two dimensional images: methodology, analysis and classification, in: 19th Int. Conf. on Control Systems and Computer Science, Bucharest, Romania, 2013, pp. 306–312.
[11] Y. Kaewaramsri, K. Woraratpanya, Improved Triangle Box-Counting Method for Fractal Dimension Estimation, in: H. Unger, P. Meesad, S. Boonkrong (Eds.), Recent Advances in Information and Communication Technology 2015, Springer International Publishing, 2015, pp. 53–61.
[12] K.C. Clarke, Computation of the fractal dimension of topographic surfaces using the triangular prism surface area method, Comput.Geosci. 12 (5) (1986) 713–722.
[13] J.J. Gagnepain, C. Roques-Carmes, Fractal approach to two-dimensional and three-dimensional surface roughness, Wear 109 (1-4) (1986) 119–126.
[14] A.K. Bisoi, J. Mishra, on calculation of fractal dimension of images, Pattern Recognit. Lett. 22 (6-7) (2001) 631–637.
[15] S. Buczkowski, S. Kyriacos, F. Nekka, L. Cartilier, The modified box-counting method: Analysis of some characteristic parameters, Pattern Recognit. 31 (4) (1998) 411–418.
[16] W. Ju, N.S.-N. Lam, An improved algorithm for computing local fractal dimension using the triangular prism method, Comput. Geosci. 35 (6) (2009) 1224–1233.