ALGORITHMS TO ESTIMATING FRACTAL DIMENSION OF TEXTURED IMAGES

Wen-Shiung Chen *Shang-Yuan Yuan* *Hungkuei Hsiao*† *Chih-Ming Hsieh*‡

Dept. of Electrical Engineering, National Chi Nan University, Taiwan, R.O.C.
†Da Yeh University, Taiwan, R.O.C.
‡Dept. of Information Management, Hsiuping Institute of Technology, Taiwan, R.O.C.

ABSTRACT

Two algorithms that can obtain more accurate estimate of the fractal dimension are proposed. One is the shifting DBC (SDBC) algorithm and the other one is the scanning BC (SBC) algorithm. It is theoretically proven that the SDBC algorithm approaches the estimated value closer to the exact fractal dimension than the DBC method. Simulation results show that the proposed algorithms consistently give more satisfactory results on textured images.

1. INTRODUCTION

Most of the objects in the real world are so complex and irregular in nature that they cannot be described by classical geometry. Fractal theory has become vastly popular in the recent decade. The complex shape description in terms of self-similarity was introduced by Mandelbrot [1]. The topological dimension of a set is always an integer while the Hausdorff dimension or fractal dimension may be a fraction. However, the Hausdorff dimension or has an elegant mathematical treatment. Besides the theoretical definition, one of the most popular definitions is the box dimension, which is an upper bound of the Hausdorff dimension [1][2]. The box dimension, sometimes referred to be the similarity dimension, is the fractional concept of choice for experimental sciences since it is more computationally manageable than the Hausdorff dimension. In general, the Hausdorff dimension and the box dimension often have the same numerical value. As a very useful measure of a fractal set, the computation of fractal dimension becomes an important issue. The existing methods such as the BC and the DBC methods, however, possess problems that they usually over-count the number of boxes so that the exact value of fractal dimension can not be obtained. Therefore, the aim of this paper is to investigate the problems and then propose two simple and efficient algorithms for obtaining the estimated value of fractal dimension closer to the exact value.

2. FRACTAL DIMENSION

Let $\mathcal{H}(\mathbb{R}^m)$ be the space of all nonempty compact subsets of the Euclidean space $\mathbb{R}^m$. A set $\mathcal{A} \in \mathcal{H}(\mathbb{R}^m)$ is said to be self-similar if $\mathcal{A}$ is the union of distinct of closed balls of radius r needed to cover $\mathcal{A}$. Let $N(\mathcal{A}, r)$ be the number of closed balls of radius r needed to cover $\mathcal{A}$. If

$$D_f(\mathcal{A}) = \lim_{r \to 0} \frac{\ln N(\mathcal{A}, r)}{\ln(1/r)} \tag{1}$$

exists, then $D_f(\mathcal{A})$ is called the *fractal dimension* of $\mathcal{A}$ and we will say "$\mathcal{A}$ has fractal dimension $D_f(\mathcal{A})$." In general, it is hard to directly compute the exact value of $D_f(\mathcal{A})$. The box-counting method is the most popular method [3]–[5].

Sarkar and Chaudhuri [6]–[8] proposed an efficient approach, called the *differential box-counting* (DBC) approach, to estimate the fractal dimension (also referred as box dimension) D_B of a 2-D gray-level image and asserted that it can precisely estimate D_B. Although the DBC method gives a better estimation of the fractal dimension, its major shortcoming is that the estimated values of $N_r(\mathcal{A})$ are not exactly the least number of boxes of side r needed to cover the 2-D fractal intensity surface. The possibility of error exists due to the effect of the quantization nature of the approach, especially when the image intensity surface is smooth. A modified approach of the DBC method with random shift of whole column of boxes in the z direction (or x/y direction) was proposed that the small variation of the gray value in image can be captured. However, the random shifting operation of columns needs additional computation involving random number generator and differential box counting in each column.

3. THE PROPOSED ALGORITHMS

A. The Shifting DBC (SDBC) Algorithm

It is easy to find out that the DBC method usually over-counts the number of boxes $n_r(i, j)$ covering the image intensity surface in the (i, j)-th grid column. The example, shown in Figure 1, demonstrates that the quantity of 3 boxes rather than 4 boxes is enough to cover the gray level variation of the intensity surface in the column of boxes if the boxes are appropriately shifted along the z direction. This results from the *quantization effect* that in the BC method the gray level variation in the z coordinate is subdivided

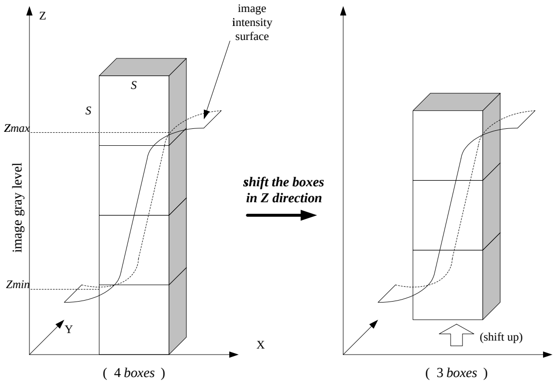


Fig. 1. The first problem of the BC and DBC methods.

into some intervals with size s'. Instead of the process of "*quantization-computation*" adopted by the DBC method, the process of "*computation-quantization*" is adopted in our approach to overcome the quantization effect. The process of "*quantization-computation*" means that before computing the number of boxes covering the intensity surface, the quantization of the z coordinate is done first. More precisely, we give the following theorem.

Theorem 1 *Let $\mathcal{A} \in \mathcal{H}(\mathbb{R}^m)$. For $m = 1, 2, 3$, we have*

$$2^{-m} N'_{2^{1-n}}(\mathcal{A}) \overset{(a)}{\leq} 2^{-m} N_{2^{1-n}}(\mathcal{A}) \overset{(b)}{\leq} N(\mathcal{A}, 2^{-n}) \overset{(c)}{\leq} N'_{2^{-n}}(\mathcal{A}) \overset{(d)}{\leq} N_{2^{-n}}(\mathcal{A}), \quad (2)$$

whenever $n \in \mathbb{N}$. Furthermore, if

$$D'_f(\mathcal{A}) = \lim_{n \to \infty} \frac{\ln N'_{2^{-n}}(\mathcal{A})}{\ln(2^{-n})} \quad (3)$$

exists, then $\mathcal{A}$ has fractal dimension $D'_f(\mathcal{A})$. Conversely, if $\mathcal{A}$ has fractal dimension $D_f(\mathcal{A})$, then $D_f(\mathcal{A}) = D'_f(\mathcal{A})$.

B. The Scanning Box-Counting (SBC) Algorithm

Unfortunately, the SDBC algorithm only can conquer the first problem but the second and the third one. In the following, for estimating more accurately than the DBC method and the SDBC algorithm a more general and efficient approach is proposed. If interpreted as a 3-D terrain whose height of the 3rd dimension is given by the pixel value, a 2-D image intensity surface may virtually be considered as a continuous surface on the 3-D space. By partitioning the continuous surface into fixed boxes, both the DBC method and the SDBC algorithm are highly possible to over-count the number of boxes. To overcome the shortcomings such as over-counting and under-counting, we propose a novel approach, called the scanning box-counting (SBC) algorithm. The fundamental principle of the SBC algorithm is counting the number of boxes by tracking the image intensity surface

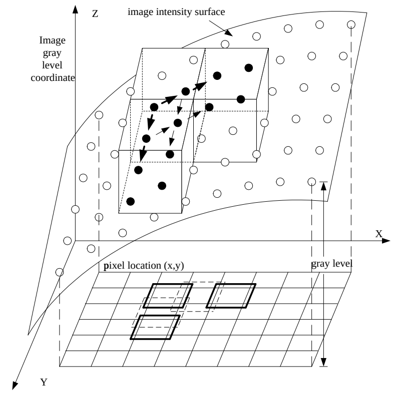


Fig. 2. The principle of the SBC algorithm.

change rather than partitioning the 3-D space into fixed boxes.

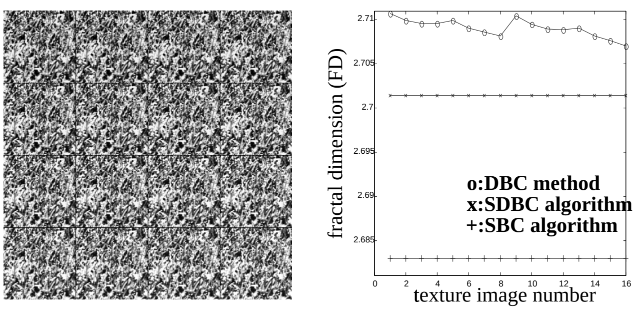
Suppose $z(m,n)$ represents the gray level of a 2-D image, where m and n are the coordinate indexes, respectively. For each pixel $z(m,n)$, two gray level differences $\Delta z_V(m,n)$ and $\Delta z_H(m,n)$ are computed, where $\Delta z_V(m,n) = z(m,n) - z(m,n-1)$ represents the difference between the pixel $z(m,n)$ and the previous neighboring pixel $z(m,n-1)$ in the vertical direction, and $\Delta z_H(m,n) = z(m,n) - z(m-1,n)$ represents the difference between the pixel $z(m,n)$ and the previous neighboring pixel $z(m-1,n)$ in the horizontal direction. One then splits $\Delta z_V(m,n)$ and $\Delta z_H(m,n)$ into $\lceil \frac{\Delta z_V(m,n)}{s} \rceil$ and $\lceil \frac{\Delta z_H(m,n)}{s} \rceil$ sub-segments in two directions.

Similarly, for each sub-segment in both two directions one checks if it has already been covered by one of the $s \times s$ previous neighboring boxes created in the scanning history. If "no," then a new box covering this sub-segment is created and counted into $N(s, \mathcal{A})$ (i.e., $N_r(\mathcal{A})$) by 1, and its corresponding location is recorded. Otherwise, no new box for this sub-segment needs to be created. After checking all of the sub-segments for the current pixel $z(m,n)$, the scanning algorithm skips to the next pixel. The procedure proceeds until all of the pixels in the computed image are processed. However, overlapping will occur if the pixel scanning order by horizontal scanning (i.e., row by row), vertical scanning (i.e., column by column), or zig-zag scanning is used. In order to avoid the overlapping problem, a special scanning order is proposed in the 2-D SBC algorithm.

4. EXPERIMENTAL RESULTS

Experiments have been performed on two sets of sixteen natural textured images taken from Brodatz's image database [9] and two sets of sixteen synthetic texture-like images. The image sizes are 128×128. The gray levels range from 0 to 255. For the first experiment, we generate a set of sixteen natural textured images with the same resolution, as shown in Figure 3(a), in which each one is simply a copy of a seed image via 16 distinct gray level shifts.

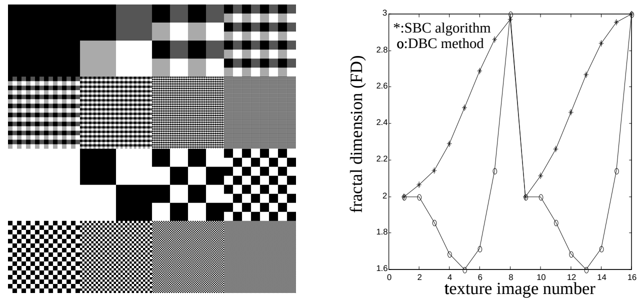
The values of fractal dimension estimated by the DBC method and the SDBC algorithm performed on the image generated above are shown in Figure 3(b). As described previously, the corresponding fractal dimensions should be the same if two intensity surfaces are entirely identical except they are with different average gray levels. We note that the estimated values of the fractal dimensions obtained by the DBC method drift along with the gray level shifting, although this contradicts the concept of fractal dimension. Conversely, the SDBC algorithm computes the same estimated values of fractal dimensions for different images. The reason is that in the SDBC algorithm the boxes used in covering the image are allowed to drift up or down along with the gray level shifting up or down. This, however, cannot be done in the DBC method. Consequently, the SDBC algorithm modified from the DBC method will estimate fractal dimension more precisely than the DBC method. Similarly, as shown in Figure 3(b), the SBC algorithm also computes the same estimated values of fractal dimensions for the same set of images shown in Figure 3(a) so that the SBC algorithm will estimate fractal dimension more precisely than the DBC method. That is, both the SDBC algorithm and the SBC algorithm reflect the roughness of an image surface more faithfully than the DBC method.



(a) (b)

Fig. 3. (a) Natural textured images with 16 distinct gray level shifts. (b) The comparison of the estimated values of fractal dimensions.

For the second experiment, we generated a set of sixteen 128×128 synthetic texture-like images, as shown in Figure 4(a). The pixel gray levels of the 1st image are all set to be “0.” The pixel gray levels of each of four sub-images (the resolutions are 64×64) in the 2nd image are “0,” “85,” “170,” and “255,” respectively. The remaining images (3rd, 4th, ..., 8th) are generated from the previous level image by down-sampling the resolution by 2 and then copying it to four quadrants. Similarly, the pixel gray levels of the 9th image are all set to be “255.” The pixel gray levels of each of four sub-images in the 10th image are “0,” “255,” “255,” and “0,” respectively. Following the same generation process as above, the remaining images (11st, 12nd, ..., 16th) are also generated.



(a) (b)

Fig. 4. (a) 16 Synthetic texture-like images. (b) The estimated values of the fractal dimensions.

The SBC method is performed to estimate the fractal dimension for each of this set of synthetic images. As Figure 4(b) shows, for those images having sharp gray level abruption just at the border of two neighboring boxes, the estimated values of fractal dimensions by the DBC method may lie outside the range between 2 and 3. This result, of course, is not reasonable. It clearly demonstrates that the DBC method cannot exactly estimate fractal dimension, thus cannot exactly manifest the roughness of an image. Conversely, the SBC algorithm considers the discrete intensity image as a continuous surface and leads the boxes tracking the surface. This box tracking process will not miss out any box at the location having thr sharp gray level abruption. As a result, the SBC method can estimate fractal dimension more exactly. Therefore, the SBC algorithm reflects the roughness of an image surface more faithfully than the DBC method for this kind of textured images. However, it has a slightly higher computational complexity than that of the DBC method. Unfortunately, the SDBC algorithm still cannot resolve the problem.

Figure 5 demonstrates the comparison of the values of the fractal dimensions estimated by the DBC method, the SDBC algorithm, and the SBC algorithm for the first set of sixteen Brodatz’s natural textured images. All of the approaches estimate the fractal dimension values very close. Two close curves reflect the very similar degree of roughness discrimination. Strictly speaking, the fractal dimension values estimated by the SDBC algorithm is a little bit smaller than that by the DBC method. The result totally coincides with the statement supported by Theorem 1. Similarly, Figure 5 also demonstrates the comparison of the values of the fractal dimensions estimated by the DBC method and the SBC algorithm for the same set of images. From the figure, the SBC algorithm obtains the least estimated fractal dimension values for the first eight natural textured images while for the remaining eight images if not so .

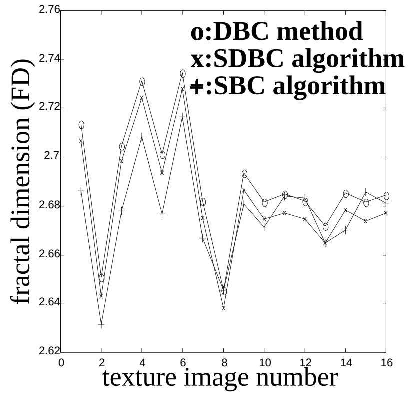
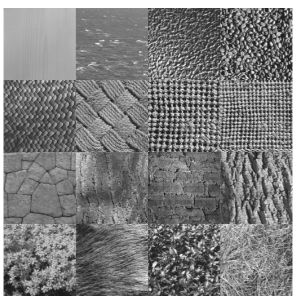
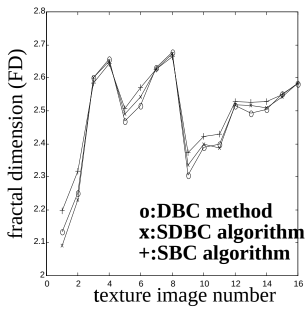


Fig. 5. The comparison of the estimated values of fractal dimensions for the first set of images.



(a)



(b)

Fig. 6. (a) Sixteen natural textured images from Brodatz database. (b) Comparison of the fractal dimensions.

Finally, the three approaches are again applied to estimate fractal dimensions for the set of sixteen distinct textured images, as shown in Figure 6(a), from Brodatz's database. The fitted error E expressed as the root mean-squared distance of the data points from the line is defined as $E = \frac{\sqrt{\sum_{i=1}^{n} \frac{(mx_i + c - y_i)^2}{(1+m^2)}}}{n}$. The error E provides a good measure of fit so that the lower the value E, the better the fit. The fractal dimensions and the errors E computed for the images shown in Figure 6(a) using different algorithms are presented in Table 1. The experimental result is also depicted in Figure 6(b). It is readily found that both the SDBC and SBC algorithm are superior to the DBC algorithm and concluded that the SBC algorithm is the best from the view point of fitted error.

5. CONCLUSION

In this paper two simple and efficient algorithms are proposed to obtain more accurate estimate of fractal dimension for 2-D textured images. As compared to the DBC method and the other kind of methods, Our algorithmsare both give more satisfactory results.

Table 1. The values of fractal dimension and fitted errors for the 2nd set of images.

	DBC		SDBC		SBC	
images	FD	E	FD	E	FD	E
1	2.133415	0.011122	2.092197	0.010120	2.198054	0.003119
2	2.250913	0.019251	2.229082	0.014112	2.316987	0.008835
3	2.599949	0.031037	2.601203	0.029028	2.583010	0.025773
4	2.655817	0.034717	2.650059	0.032335	2.643636	0.026560
5	2.470243	0.029972	2.491461	0.028589	2.507317	0.021436
6	2.516046	0.026402	2.542373	0.022635	2.571014	0.016201
7	2.631907	0.037651	2.627138	0.034498	2.627080	0.026294
8	2.677373	0.031890	2.672149	0.030420	2.665015	0.024411
9	2.305452	0.020534	2.336494	0.021650	2.371888	0.013025
10	2.390279	0.026098	2.398498	0.021640	2.422303	0.015371
11	2.399804	0.025913	2.386965	0.021676	2.430104	0.013988
12	2.516629	0.033771	2.517884	0.031688	2.527595	0.024141
13	2.492698	0.031454	2.517179	0.028788	2.526153	0.020914
14	2.504440	0.021431	2.509932	0.018509	2.527374	0.014425
15	2.551598	0.034458	2.540916	0.034036	2.548192	0.025639
16	2.581626	0.026041	2.583370	0.023330	2.585091	0.019299
average	error	0.027608		0.025190		0.018714

1. REFERENCES

[1] B. Mandelbrot, *Fractal Geometry of Nature*. San Francisco: Freeman, 1982.

[2] M. F. Barnsley, *Fractal Everywhere*, 2nd Edition. Academic Press Professional, 1993.

[3] J. M. Keller, R. M. Crownover and S. S. Chen, "Characteristics of natural scenes related to the fractal dimension," *IEEE Trans. on Pattern Analysis and Machine Intelligence,* vol. PAMI-9, No. 5, pp. 621-627, Sep. 1987.

[4] J. M. Keller, S. S. Chen and R. M. Crownover, "Texture description and segmentation through fractal geometry," *Comput. Vision, Graphics and Image Processing,* vol. 45, pp. 150-166, 1989.

[5] S. S. Chen, M. Keller and R. M. Crownover, "On the calculation of fractal features from images," *IEEE Trans. on Pattern Analysis and Machine Intelligence,* vol. PAMI-15, No. 10, pp. 1087-1090, Oct. 1993.

[6] N. Sarkar and B. B. Chaudhuri, "An efficient approach to estimate fractal dimension of textured images," *Pattern Recognition,* vol. 25, No. 9, pp. 1035-1041, 1992.

[7] N. Sarkar and B. B. Chaudhuri, "An efficient differential box-counting approach to compute fractal dimension of image," *IEEE Trans. on Systems, Man, and Cybernetics,* vol. 24, No. 1, pp. 115-120, Jan., 1994.

[8] B. B. Chaudhuri and N. Sarkar, "Texture segmentation using fractal dimension," *IEEE Trans. on Pattern Analysis and Machine Intelligence,* vol. 17, No. 1, pp. 72-77, Jan. 1995.

[9] P. Brodatz, *Texture: A Photographic Album for Artists and Designers.* New York: Dover, 1966.