Pattern Recognition Letters 19 (1998) 309–313

Pattern Recognition Letters

Fractal dimension estimation for texture images: A parallel approach

Manoj Kumar Biswas [a], Tirthankar Ghose [b], Sudipta Guha [b], Prabir Kumar Biswas [a,*]

[a] *Department of Electronics and Electrical Communication Engineering, Indian Institute of Technology, Kharagpur 721 302, India*
[b] *Department of Mathematics, Indian Institute of Technology, Kharagpur 721 302, India*

Received 3 September 1996; revised 19 December 1997

Abstract

Fractal dimension is an important parameter that can be used in various applications, such as, estimation of roughness in an image, texture segmentation, surface roughness estimation and many others. A number of techniques for fractal dimension computation in the digital domain have been reported in the literature. A parallel implementation of the Differential Box Counting technique is reported in this paper. The accuracy and computational complexity of the parallel implementation are also discussed.

Keywords: Fractal dimension; Differential Box Counting; Parallel algorithm; SIMD array processor

1. Introduction

Geometric primitives which are self-similar and irregular in nature are called fractals. Most of the natural objects, at the macro-level as well as at the micro-level should better be called fractal objects rather than a combination of Euclidean primitives. That is why fractal analysis is preferred in most of the applications where natural objects are dealt with. Fractal dimension can be used as a measure of complexity or irregularity of a curve or a surface, and therein lies its importance for applications. Fractal analysis has found wide applications in areas ranging from material science (Pfeir, 1984), geology (Goodchild, 1980), power technology (Clark, 1986), computer vision (Pentland, 1984; Hartley et al., 1984; Chaudhuri and Sarkar, 1995) to micro-electronics (Spanos and Irene, 1994). In most of these applications, the common interest is to determine the fractal dimension of the concerned objects.

A number of techniques for estimation of fractal dimension have been reported in the literature. Pentland (1984) developed a three-dimensional fractal model for image segmentation and estimated the fractal dimension using the Fourier power spectral density where the surfaces were modeled as fractional Brownian Motion (fBM) surfaces. Hartley et al. (1984) used the ε-blanket method suggested by Mandelbrot (1982) to estimate the fractal dimension and used it for texture analysis. Dubuc et al. (1989)

* Corresponding author. Email: pkb@ece.iitkgp.ernet.in.

0167-8655/98/$19.00
PII S0167-8655(98)00002-6

used the variation method for fractal dimension estimation. Spanos and Irene (1994) used the two-dimensional variation method to estimate the surface roughness of silicon surfaces. Sarkar and Chaudhuri (1992, 1994) introduced the Differential Box Counting (DBC) method and used it for texture image segmentation (Chaudhuri and Sarkar, 1995). They concluded the following:

1. The DBC ranks as the best in terms of computational efficiency.
2. The methods proposed by Peleg et al. and Pentland ranks as the best in terms of accuracy.
3. The DBC and the method proposed by Pentland rank as the best in terms of dynamic range of fractal dimension.

Jin and Ong (1995) suggested an improvement in the scale limit of the DBC method in their Relative Differential Box Counting (RDBC) technique.

In this paper we suggest a modified implementation of the Differential Box Counting technique (Sarkar and Chaudhuri, 1992, 1994) to make it computationally more efficient. We also propose a parallel implementation of the modified technique on a shared memory array processor.

2. About fractal dimension

Mandelbrot (1982) defined a fractal as a bounded set A in $\mathbb{R}$ for which the Hausdorff–Besicovich (HB) dimension is strictly larger than the topological dimension, where HB or fractal dimension is a real number used to characterize the geometric complexity of A. The concept of self-similarity can be best utilized to estimate the fractal dimension. A bounded set A in Euclidean n-space is self-similar if A is the union of N_r distinct (non-overlapping) copies of itself scaled up or down by a factor of r. The fractal dimension D of A is given by relation (1).

$$1 = N_r r^D. \tag{1}$$

$$D = \frac{\log N_r}{\log\left(\dfrac{1}{r}\right)}. \tag{2}$$

The Differential Box Counting method for fractal dimension computation considers an image of size $M \times M$ which is scaled down to $s \times s$ grids, where $2 \leqslant s \leqslant \frac{1}{2}M$. Then $r = s/M$. Each grid can be viewed as a column of boxes of size $s \times s \times s'$ placed one above the other, where s' satisfies the relation $\lfloor M/s \rfloor = \lfloor G/s' \rfloor$, G being the total number of gray levels. If for the (i,j)th grid, maximum and minimum gray levels of the image lie in the kth and lth boxes respectively, then $n_r(i,j) = k - l + 1$ is the contribution of N_r to the (i,j)th grid.

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

From expression (2), using the least square error linear fit for $\log N_r$ against $\log(1/r)$, we obtain the fractal dimension as the slope of the fitted line. So the major step in fractal dimension estimation is the calculation of N_r for different values of r.

3. Proposed algorithm

The range of grid sizes, proposed by Sarkar and Chaudhuri (1992, 1994) is $2 \leqslant s \leqslant \frac{1}{2}M$, whereas we consider the grid sizes as $s = 2^i$, where i is an integer. Since our proposed set of grid sizes is a subset of the set of all possible grid sizes, the fitted straight line will be approximately identical. Selection of grid sizes as a power of 2 enables us to implement the DBC technique in a more computationally efficient manner, which avoids scanning of the whole image again and again. Moreover this particular sequential algorithm can be converted into a parallel one which can be directly implemented on a shared memory SIMD array processor.

3.1. Sequential algorithm

Algorithm 1.
Aim. To calculate N_r.
Input. Image array I of size $M \times M$ is copied into two buffers $I^{\max}$ and $I^{\min}$.
Output. Sequence of N_r for various values of r ($r = s/M$).
1. $s \leftarrow 2$
2. $size \leftarrow M$
3. while $size > 2$ do
begin

3.1. Calculate s' using $\lceil M/s \rceil = \lceil G/s' \rceil$
3.2. $N_r \leftarrow 0$
3.3. for $i = 0$ to $size - 1$ in steps of 2
  begin
  3.3.1. $sum \leftarrow 0$
  3.3.2. for $j = 0$ to $size - 1$ in steps of 2
    begin
    3.3.2.1. $I^{\max}_{i/2,j/2} \leftarrow$
      $\max(I^{\max}_{i,j}, I^{\max}_{i,j+1}, I^{\max}_{i+1,j}, I^{\max}_{i+1,j+1})$
    3.3.2.2. $I^{\min}_{i/2,j/2} \leftarrow$
      $\min(I^{\min}_{i,j}, I^{\min}_{i,j+1}, I^{\min}_{i+1,j}, I^{\min}_{i+1,j+1})$
    3.3.2.3. $sum \leftarrow sum +$
      $\lceil (I^{\max}_{i/2,j/2})/s' \rceil - \lceil (I^{\min}_{i/2,j/2})/s' \rceil + 1$
    end
  3.3.3. $N_r \leftarrow N_r + sum$
  end
3.4. Save N_r and $r = s/M$
3.5. $s \leftarrow s \times 2$
3.6. $size \leftarrow size/2$
end

In the ith iteration, the number of data points under operation is $N/4^{i-1}$, where N is the total number of image points $(N = M \times M)$. The loop is executed $\log(\frac{1}{2}M)$ times. So the total number of image points considered in this algorithm is $N\sum_{i=1}^{\log(M/2)}(\frac{1}{4})^{i-1}$, which is approximately $1.33N$.

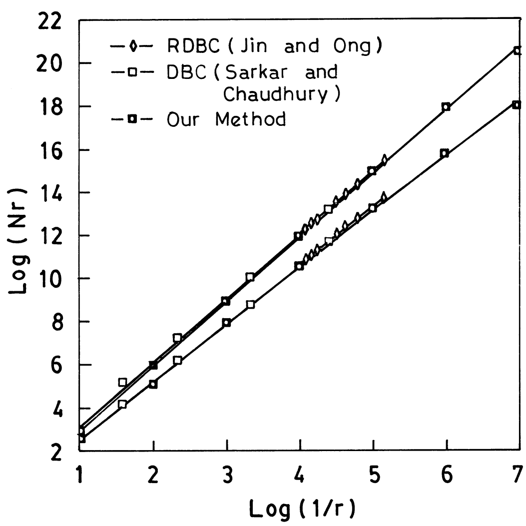

Fig. 1. log N_r versus $\log(1/r)$ for two texture images.

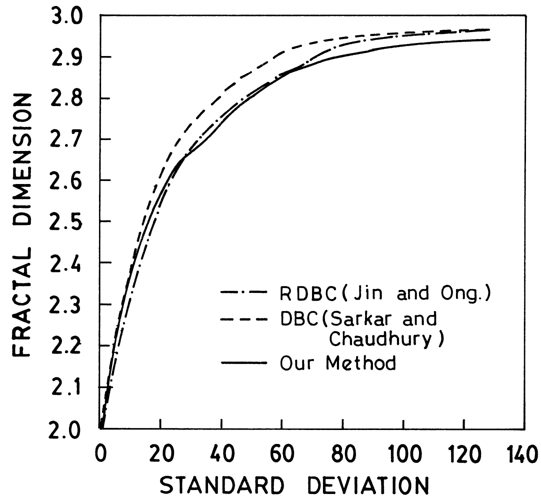

Fig. 2. Estimated fractal dimension versus standard division of noise on simulated plane images using various methods.

The direct implementation requires operation on all N data points for every grid size. Thus, if there are m discrete grid sizes, the total number of image points to be considered is $m \times N$. Usually m is much larger than 1.33. Hence, the number of operations required in the modified technique is much less than that in the direct technique.

We have implemented the DBC technique (Sarkar and Chaudhuri, 1992, 1994) with different sets of grid sizes including our proposed one and the set proposed by Jin and Ong (1995) to observe the effect of choice of different grid sizes on fractal dimension estimation. The results are depicted in Fig. 1. These results show that $\log(N_r)$ versus $\log(1/r)$ plots in all cases are nearly identical, which in turn supports our claim that the choice of the grid size has little influence on fractal dimension estimation. The range of fractal dimension against the standard deviation of white Gaussian noise on a simulated plane image is presented in Fig. 2. From the plot it is observed that the range obtained using our technique is close to the one obtained using the DBC technique.

3.2. Parallel algorithm

The modified DBC technique is mapped onto a shared memory SIMD array processor. Each image pixel $I(i,j)$ is stored into two locations in the shared memory namely $p_{i,j}^{\max}$ and $p_{i,j}^{\min}$ and associated with the (i,j)th processing element of the array processor. The algorithm is presented below.

Algorithm 2.
Aim. To calculate N_r.
Input. Image data stored in $p_{i,j}^{\max}$ and $p_{i,j}^{\min}$ which are linked with the (i,j)th processing element. $p_{i,j}^{\max}$ and $p_{i,j}^{\min}$ will have the respective maximum and minimum values.
Output. Sequence of N_r for various values of r.
1. $s \leftarrow 2$
2. while $s < \frac{1}{2}M$
begin
2.1. Calculate s' from $\lceil G/s \rceil = \lceil M/s \rceil$
2.2. $N_r \leftarrow 0$
2.3. $\forall i,j < M/s$ do in parallel
begin
2.3.1. $p_{s\times i,s\times j}^{\max} \leftarrow \max(p_{s\times i,s\times j}^{\max}, p_{s\times i,s\times j+s/2}^{\max})$
2.3.2. $p_{s\times i,s\times j}^{\min} \leftarrow \min(p_{s\times i,s\times j}^{\min}, p_{s\times i,s\times j+s/2}^{\min})$
end
2.4. $\forall i,j < M/s$ do in parallel
begin
2.4.1. $p_{s\times i,s\times j}^{\max} \leftarrow \max(p_{s\times i,s\times j}^{\max}, p_{s\times i+s/2,s\times j}^{\max})$
2.4.2. $p_{s\times i,s\times j}^{\min} \leftarrow \min(p_{s\times i,s\times j}^{\min}, p_{s\times i+s/2,s\times j}^{\min})$
end
2.5. $\forall i,j < M/s$ do in parallel
$n_r(s\times i, s\times j) \leftarrow \lceil p_{s\times i,s\times j}^{\max}/s' \rceil - \lceil p_{s\times i,s\times j}^{\min}/s' \rceil + 1$
2.6. $N_r \leftarrow \sum_{i,j<M/s} n_r(s\times i, s\times j)$ in parallel
2.7. Save N_r and $r = s/M$
2.8. $s \leftarrow s \times 2$
end

In the parallel algorithm, the ''while'' loop starting at step 2 is executed $\log(\frac{1}{2}M)$ times, where $M \times M$ is the image size. Steps 2.1–2.5, 2.7 and 2.8 take constant time to be executed in parallel. The time complexity of step 2.6 is $O(\log n)$, where $n = N/4^{i-1}$ in the ith iteration of the ''while'' loop. Hence the time complexity of the parallel algorithm is $\sum_{i=1}^{\log(M/2)} \log(N/4^{i-1})$, i.e., $\Theta((\log N)^2)$, which is much less compared to that of the sequential algorithm.

4. Conclusion

In this paper we have presented a modified implementation of the Differential Box Counting technique for fractal dimension computation. We have studied the accuracy of the modified technique and found that the original DBC technique and our implementation give more or less the same result. Though time complexity of both the algorithms are of the same order, the number of operations in our implementation is much less compared to that in the original DBC technique. The advantage of the proposed modification is that it can be easily parallelized and an implementation on a shared memory array processor is also suggested. The parallel algorithm is found to be computationally much more efficient than the sequential version.

Acknowledgements

The authors are grateful to the anonymous reviewers for their valuable comments on an earlier version of this paper.

References

Chaudhuri, B.B., Sarkar, N., 1995. Texture segmentation using fractal dimension. IEEE Trans. Pattern Anal. Machine Intell. 17, 72–77.
Clark, N.N., 1986. Power Tech. 46, 45.
Dubuc, B., et al., 1989. Evaluating the fractal dimension of profiles. Physics Rev. A 39, 1500–1512.
Goodchild, M.F., 1980. Math. Geo. 12, 85.
Hartley, R., Peleg, S., Naor, J., Avnir, D., 1984. Multiple resolution texture analysis and classification. IEEE Trans. Pattern Anal. Machine Intell. 6, 518–523.
Jin, X.C., Ong, S.H., 1995. A practical method for estimating fractal dimension. Pattern Recognition Lett. 16, 457–464.
Mandelbrot, B.B., 1982. Fractal Geometry of Nature. Freeman, San Francisco.
Pentland, A.P., 1984. Fractal based description of natural scenes. IEEE Trans. Pattern Anal. Machine Intell. 6, 661–674.

Pfeir, P., 1984. Appl. Surf. Sci. 18, 146.

Sarkar, N., Chaudhuri, B.B., 1992. An efficient approach to estimate fractal dimension of textural images. Pattern Recognition 25, 1035–1042.

Sarkar, N., Chaudhuri, B.B., 1994. An efficient differential box counting approach to compute fractal dimension of images. IEEE Trans. Systems Man Cybernet. 24, 115–120.

Spanos, L., Irene, E.A., 1994. Investigation of the roughened silicon surfaces using fractal analysis, Part I: Two dimensional variation method. J. Vac. Sci. Technol. A 12, 2646–2652.