Complex Systems 1 (1987) 401–416

Closed Loop Identification in Diffusion-limited Aggregation Processes

Jack C. H. Chung
Maria Litt
Gary G. Leininger
Harvey Scher
The Standard Oil Company, Research and Development Center, Information Sciences Laboratory, 4440 Warrensville Center Road, Cleveland, OH 44128, USA

Abstract. Diffusion-limited aggregation (DLA) is a useful model for studying such common physical phenomena as dust clustering, unstable fluid flow, chemical species precipitation, and crystal growth. Simulating the DLA processes using electrostatic analogy is very computation-intensive. Elimination of closed loops in DLA images can significantly reduce the dimensionality of the problem and minimize the computational time required. In this paper, we describe both non-recursive and recursive techniques for the identification of closed loops in DLA processes. The recursive algorithm developed can identify closed loops without rescanning the complete image at each stage of the aggregation process. Computer simulations indicate that the recursive algorithm is several orders more efficient than the non-recursive one.

1. Introduction

Diffusion-limited aggregation (DLA) is an idealized representation of a variety of common unstable kinetic growth processes such as dust clustering [1], chemical species precipitation from a supersaturated matrix, and crystal growing from a supercooled melt [2].

Witten and Sander [3,4] used a random walk process to simulate such diffusion-limited aggregation. They start with a seed particle at the origin of a lattice, then another particle is generated at a far away location and allowed to walk at random until it reaches one of the lattice sites adjacent to the occupied site, called stick points (or perimeter sites). The particle then sticks to the cluster and creates more stick points; another particle is launched and the random walk process is repeated. An indefinitely large cluster can be formed this way. A typical structure produced on a two-dimensional lattice is shown in figure 1 [3].

© 1987 Complex Systems Publications, Inc.

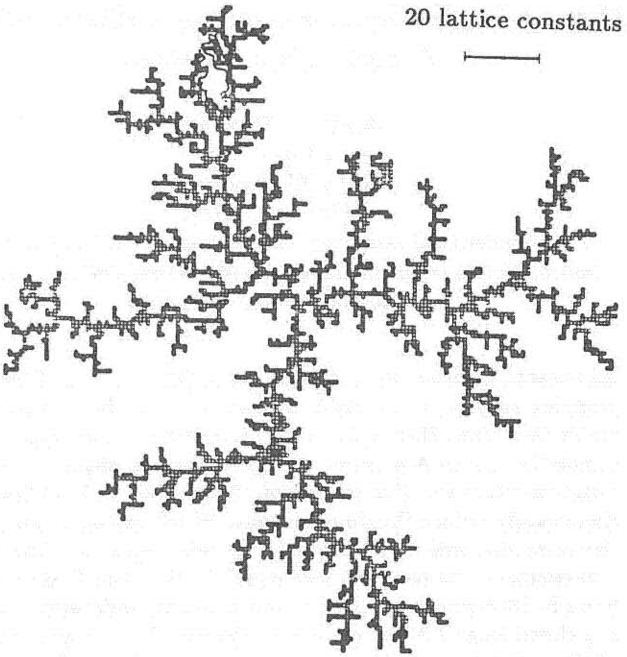

Figure 1: A typical DLA image (from [3], courtesy of Dr. L. Sander and the American Physical Society).

There is one drawback with simulating the random walk process directly: the random walkers may walk around the cluster and take a long time to reach any stick point, or they may even walk completely out of the lattice boundary. Instead, the electrostatic analogy can be used to calculate the probabilities of occupancy of each stick point and new particles can be generated directly according to these probabilities [6]. The calculation of the probabilities involves the inversion of a large matrix whose dimension is proportional to the number of stick points. During the evolution of the cluster, closed loops may be formed causing the stick points inside the loops to become inaccessible. Elimination of these inaccessible stick points from probability calculations can significantly reduce the dimensionality of the problem and minimize the computational time required.

In this paper, we will describe image processing techniques for the identification of closed loops in DLA images. In section 2, a non-recursive technique is described which is a modification of the component labeling technique used in image processing. In section 3, a recursive algorithm is developed specifically for the DLA images. The comparison of the non-recursive and recursive algorithms is given in section 4.

2. Non-recursive closed loop identification

One way to identify closed loops in DLA images is to use the component labeling technique [5] of image processing by treating each closed loop as a separate component. The basic procedure of this algorithm works as follows:

1. For any given image, label each unoccupied site (pixel) by scanning the image line by line. If neither the left nor the upper neighbor is an unoccupied site (not a stick point), then give the current site a new label. If either the left or the upper neighbor is an unoccupied site, then give the current site that neighbor's label. Otherwise, if both neighbors are unoccupied sites, then give the current site either neighbor's label and take note that both neighbor's labels are equivalent.

2. After labeling the complete image, convert all equivalent labels to minimum set required.

3. Those unoccupied sites having the same label as the image boundary points are accessible sites, while those with labels different from that of the image boundary points are closed loops and inaccessible from outside.

Figure 3 shows the result of applying the modified component labeling technique to the image in figure 2; the inaccessible sites are marked as Xs.

```
                      0
                    0 00
                  0 00000
                  000   0
                    00  0 0 00
                 00 0   000 0      0
                00 000000  000000000
                 000    0  00
               0        0 000
               0        0 000
               000  00000 00      000
      00 00     0000 0   00 0     00
     0 0  0       000000000 0  0000
     000000        000 00000 00 0000
       00000 0     0000  0     0 0 0
       00 00000    0 0000000   000 00
       000    00   0000 0  0 000 00000
     0 00000   0000 00  000000 00000000
     0   00000  0 00000000000 000     0
   0000000 00000000  000000 0
   0      0000   00      0
             00000000 0000         00
     0       0  00 0 00 0000      00
    000  0     0 0 0  0 00  00 0000
00 000000000000000    00000  000000
 0000    00000  00000  0000 000  0
 0000   000000 0    00 00   00  000
  0 0 000   0 00    0000000 000   0
    0 0000 00         0 00 00   000
      0 00  0      0 00  0  0  00 0
      0 00 000      0000 0  00
     00              0 00
      0
```

Figure 2: Image before closed loop identification.

```
                       .
                     ..0.
                   ..0X00.
                  .0X00000.
                  .000XXX0X. ..
                  .XX00XX0X0.00.    .
                 .00X0XXX000X0......0.
                .00X000000XX000000000.
                .X000XXXX0XX00.......
               .0XXXXXXXX0X000.
               .0XXXXXXXX0X000.    ...
       .. ..   .000XX00000X00.    .000.
      .00.00.   .0000X0XXX00X0. ...00.
     .0X0XX0.    ..000000000X0X.0000.
     .000000X.    .X000X00000X00X0000.
      .X00000X0.   .0000XX0XXXXX0X0X0.
       .00X00000.  .0X0000000XXX000X00.
      .X000XXXX00..X0000X0XX0X000X00000.
     .0X00000XXX0000X00XX000000X00000000.
    .X0XXX00000XX0X00000000000X000.....0.
   .0000000X00000000XX000000X0....
   .0XXXXX0000XXX00XXXXXXX0.        ..
    .XXXXXXXXX00000000X0000X.      .00.
     .0XXXXXXX0XX00X0X00X0000.. ...00.
 .. .000XX0XXXXX0X0X0XX0X00XX00.0000.
.00.000000000000000XXXX00000XX000000.
 .0000XXXX00000XX00000XX0000X000XX0.
 .0000XXX000000X0X..X00X00XXX00XX000.
  .0X0X000XXX0X00.  .0000000X000XXX0.
   ..0X0000X00XX.    .XX0X00X00XXX000.
     .X0X00XX0X.    .0X00XX0XX0XX00.0.
      .0X00.000.    .0000X0..00... .
     .00X.. ...      .0.00.  ..
      .0.             . ..
       .
```

Figure 3: Image after closed loop identification.

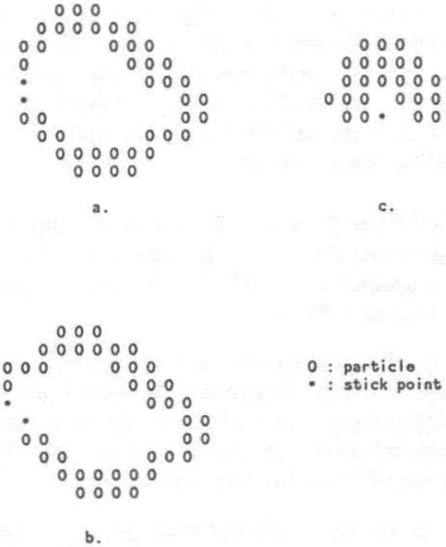

Figure 4: Formation of inaccessible closed loop.

3. Recursive closed loop identification

In a diffusion-limited aggregation process, the aggregation of particles is sequential. Since each new particle usually only slightly changes the structure of the overall image, scanning the complete image at each stage to identify closed loops is a workable yet inefficient method. Instead, it is much more economical from a computational point of view to develop a recursive algorithm for identifying inaccessible closed loops in the image. The recursive algorithm should utilize as much information from previous stages as possible in order to minimize additional computation.

3.1 Closed loop formation

In order to develop the recursive algorithm, it is necessary to understand some fundamental properties of DLA processes and associated closed loop formation. The most important property of a DLA image is that the particles always form a connected aggregate, which implies that any two particles, no matter what relative positions they occupy, are connected in some way.

In general, a closed loop is formed when the loop opening is closed by two side-by-side or diagonally placed stick points as shown in figure 4a and b. The only exception is the case shown in figure 4c where a single stick point also forms a closed loop.

Figures 5a through 5i list all possible configurations of two side-by-side stick points for one orientation. Each configuration can be rotated 90, 180, and 270 degrees to obtain different variations. In the figures, the new stick point is, by definition, a point which was a free unoccupied site and which becomes a stick point as a result of the new aggregated particle. Based on the properties of DLA process and closed loop formation discussed above, the following observations can be made:

1. Cases shown in figures 5a and i, 5b and h, 5c and g, and 5d and f are mirror image pairs; that is, one image can be obtained by flipping the other image around the vertical axis. These mirror image pairs possess similar characteristics.

2. Case in figure 5e will form a closed loop. This is due to the fact that any two particles in a DLA image are connected one way or another and the two stick points shown will close the loop opening. It is not evident, however, whether the loop is on the right (figure 6a) or on the left (figure 6b) without further processing.

3. Case in figure 5b will cause the old stick point on the left hand side of the new stick point to become inaccessible. This can be reasoned as follows. Since each particle will stick to the aggregate only when it occupies a stick point, the site where the new particle lies must have been a stick point and one of its four neighbors (east, west, south, and north) must have been an occupied site as illustrated in figures 7a through c. In either case, a closed loop was formed. The closed loop cannot be on the right side as this would have caused the site of the new stick point in figure 5b to become inaccessible, which is in contradiction to the existence of the new stick point under discussion. Therefore, the closed loop must be on the left with the point to the left of the new stick point sitting on the loop boundary. The addition of the new stick point simply expands the existing loop by enclosing the point to the left of the new stick point inside the loop. Similar results can be obtained for figures 5c, g, and h.

4. No new closed loops will be formed by the addition of the new stick point in figures 5a, d, f, and i.

Similarly, figures 8a through h list all possible configurations of two diagonally placed stick points for one specific orientation. Other variations can be obtained by rotating these images 90, 180, and 270 degrees. The following conclusions can be drawn for these cases.

1. New closed loops will definitely be formed by the addition of the new stick point in figures 8b, c, f, and g, as the two diagonally placed stick points will close the loop opening of the two locally separated but globally connected particles.

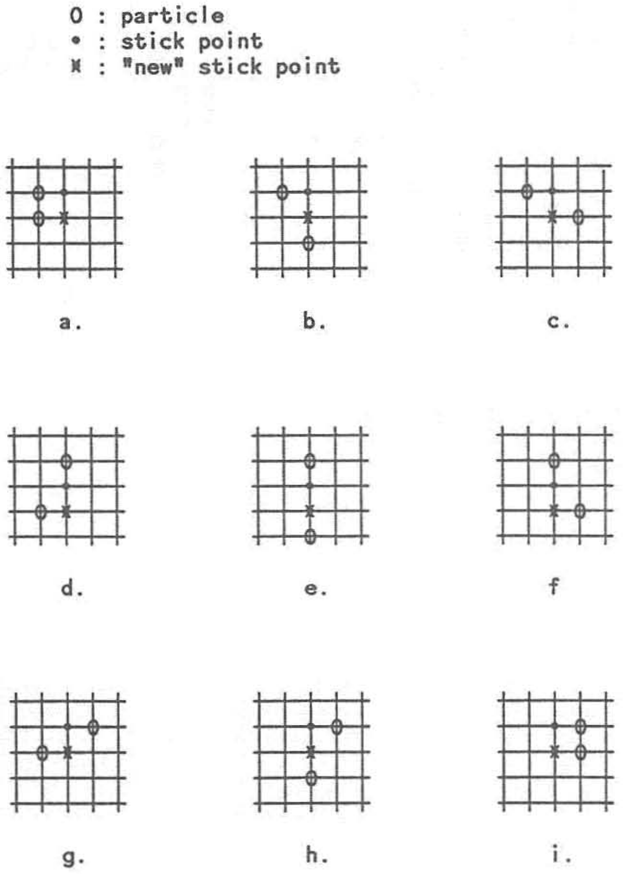

Figure 5: Different cases for two in-line stick points.

```
    0 0 0
  0 0 0 0 0 0
0 0       0 0 0
0           0 0 0
•             0 0 0
•                 0 0
0 0               0 0
  0 0         0 0 0
    0 0 0 0 0 0
      0 0 0 0

         a.
```

```
      0 0 0
    0 0 0 0 0 0
  0 0       0 0 0 0
0 0             0
0 0             •
0 0 0           •
    0 0 0 0 0 0 0
      0 0 0 0 0
        0 0 0
          0

           b.
```

Figure 6: Different loop formation.

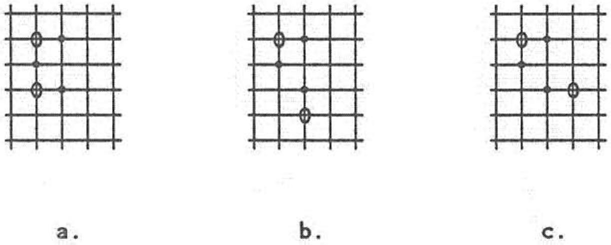

Figure 7: Possible predecessor images of figure 5b.

2. No new closed loops will be formed by the addition of the new stick point in figures 8a, d, e, and h.

3.2 Loop tracing and verification

From the above analysis, it can be concluded that existence of new closed loops is a local property which can be determined from a few neighborhood points, whereas the extent of the closed loops is a global property which can not be determined from neighborhood points. As illustrated in figure 6, the loop may be on the right or on the left, and the loop may be small or large.

One way of determining the location and extent of a closed loop is to trace the loop boundary that encloses the inaccessible sites. The loop can be traced either along the boundary particle sites or boundary stick point. Tracing along the boundary stick point is found to be more advantageous for subsequent processing and thus will be used in the algorithm. As shown in figure 9, two kinds of loops need to be differentiated: one is the real internal closed loop of the inaccessible sites; the other one is the external boundary loop surrounding the entire DLA image.

To determine whether an internal closed loop or an external boundary has been traced, notice that if during tracing a consistent convention is maintained to keep the particle sites on the right, then an internal closed loop would be traced in a counter-clockwise fashion, while an external boundary would be traced in a clockwise fashion no matter how complicated the loops are. Figure 10 shows a method to ensure that the particle sites are on the right-hand side during the tracing. At each point during the tracing, the eight neighbors of the current point and the relative positions of the current point and the previously traced point are used to determine which point should be traced next. Starting from the second clockwise point from the previously traced point, check in a clockwise fashion until a particle site or an inaccessible site is found, then back up one step and locate the point which should be traced next. The first clockwise point from the previously traced point should not be checked in order to prevent tracing back to the previous point at the start of the loop tracing.

Whether a loop traced is clockwise or counter-clockwise can be determined by calculating the total angle of directional change during tracing. A counter-clockwise loop has a total angle of +360 degrees (counter-clockwise angle is taken to be positive), whereas a clockwise loop has a total angle of -360 degrees. Figure 11 illustrates the two different cases.

The methods described above are general and can be applied to multiple object images. A heuristic method, however, is available to significantly reduce the unnecessary effort in tracing the external boundary in a DLA image. Since each DLA image contains only one single connected object, when the external boundary is traced, the stick points traced will be outside the object boundary. Therefore, if the coordinates of the four extreme

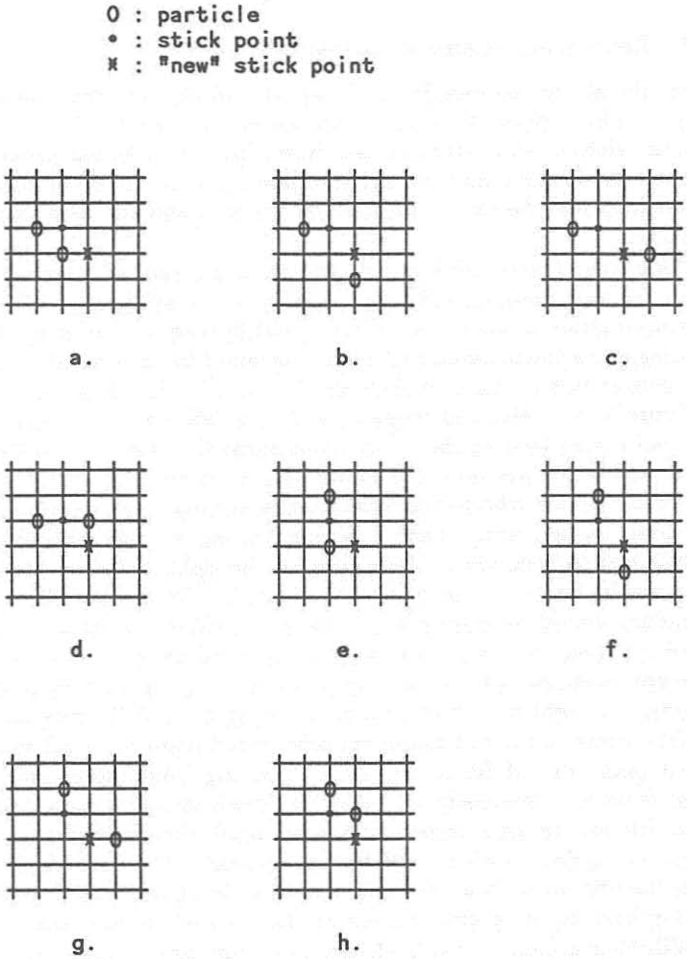

Figure 8: Different cases for two diagonally-placed stick points.

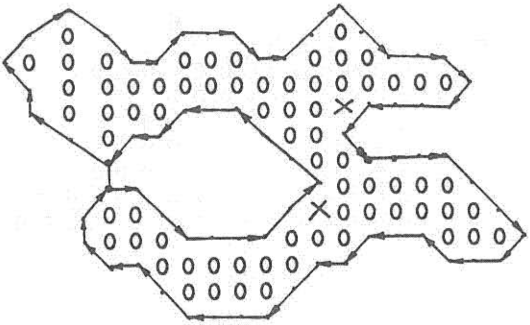

Figure 9: Internal vs. external loops.

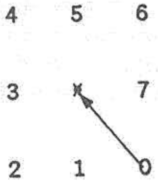

Figure 10: Convention used in closed loop tracing.

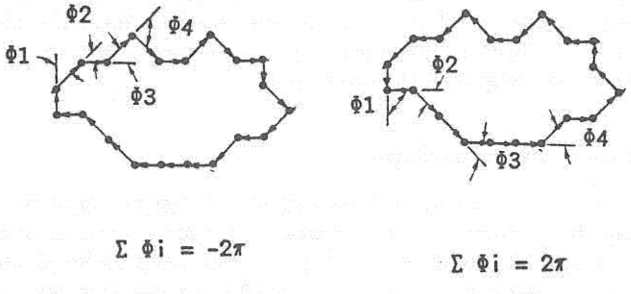

Figure 11: Clockwise vs. counter-clockwise loops.

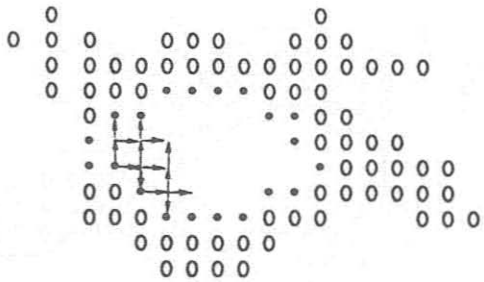

Figure 12: Region growing.

points (east, west, south, and north) of the DLA aggregate are stored during the aggregation process, tracing along the external boundary can be terminated whenever the stick point traced falls outside of the rectangle bounded by the four extreme points. By reversing the sequence of the two stick points that close the loop, the internal closed loop can be found. Using this heuristic method, successful completion of the tracing always indicates the existence and location of an internal closed loop.

3.3 Region growing

After the tracing, the location of the closed loop can be identified. The next step is to update all sites inside the closed loop into inaccessible points. To do so, the region growing technique [6] of image processing can be utilized. As shown in figure 12, we can start with any seed point inside the closed loop and update its four neighbors if necessary, then take one of the updated neighbors at a time and continue the process until all sites are updated. The growing process should stop in a particular direction when a particle site, an existing inaccessible point, or the two stick points which form the closed loop are encountered. The two stick points which form the closed loop should not be updated to inaccessible points because they are on the boundary of the closed loop and the open space.

3.4 Overall recursive algorithm

To sum up the recursive algorithm described above, figure 13a shows the complete neighborhood points of a new particle that may cause the formation of a closed loop. For each new stick point, the neighborhood points which need to be considered are shown in figure 13b. The sequence of points handled is of importance; the procedure given below shows the proper order.

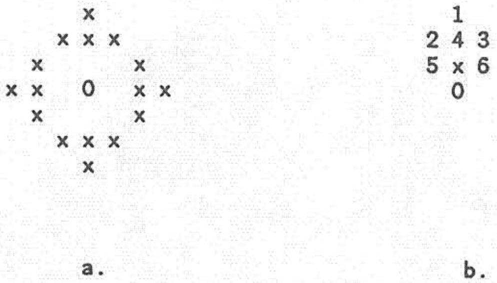

Figure 13: (a) complete neighborhood points of a new particle which may cause the formation of a closed loop; (b) the neighborhood points which need to be considered for each new stick point.

```
for (each "new" stick point generated by the new particle) do
    if (point 1 = occupied point) then do
            1. trace the closed loop formed by the two stick points 4 and $x$
               and update the image by region growing;
            2. if (point 4 = inaccessible) then update point 4;
            3. skip the rest of the program and proceed with another new
               stick point.
    end if

    if (point 2 = occupied point) then do
            set point 5 = inaccessible point;
    else if (point 2 = stick point) then do
            trace the closed loop formed by the two stick points 2 and $x$
            and update the image by region growing;
    end if

    if (point 3 = occupied point) then do
            set point 6 = inaccessible point;
    else if (point 3 = stick point) then do
            trace the closed loop formed by the two stick points 3 and $x$
            and update the image by region growing;
    end if

proceed with the next new stick point;
```

The complete recursive algorithm is coded in FORTRAN for comparison with the non-recursive algorithm. In addition, since the LISP programming language can handle recursion better than FORTRAN, the recursive algorithm has also been coded and tested in LISP. Figure 14 shows an il-

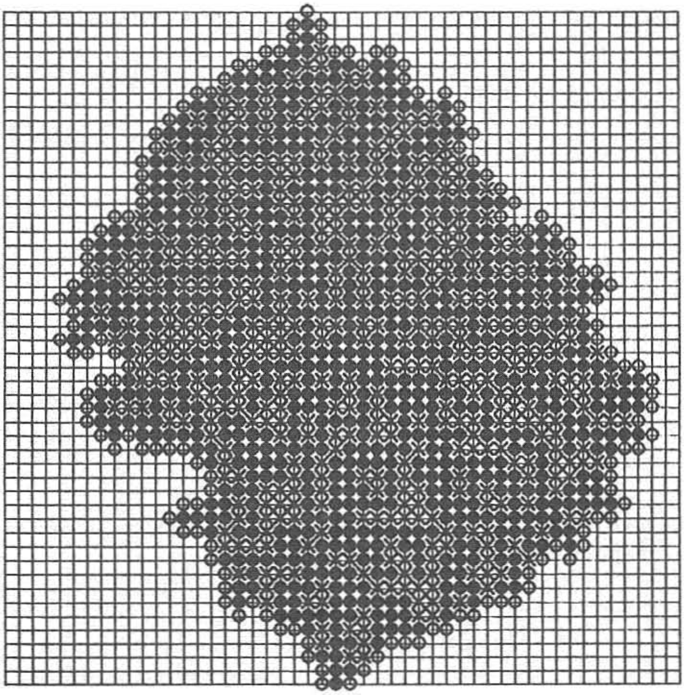

Figure 14: Recursive closed loop identification.

lustrative example of the recursive algorithm, where the occupied sites are indicated by filled circles, stick points by unfilled circles, and inaccessible points by Xs.

4. Comparison of non-recursive and recursive algorithms

The non-recursive algorithm described in section 2 and the recursive algorithm described in section 3 both produce the same result. The computation involved, however, is quite different. As each new particle sticks to the cluster, the modified component labeling technique needs to rescan the image to create the new component labels and subsequently reduce these labels to the minimum set required. Therefore, its computation is at least proportional to the square of the dimension of the image. The recursive algorithm, on the other hand, usually requires scanning of only a few neighborhood points which is independent of the size of the image; tracing and region growing are required only when closed loops occur.

To compare the recursive and non-recursive algorithms, the two algo-

Number of Particles	CPU time Recursive (second)	CPU time Non-recursive (second)	Ratio
10	0.12	2.50	1:21
100	0.14	24.89	1:178
250	0.24	60.56	1:252
500	0.37	122.53	1:331
1000	0.90	260.66	1:289
1500	1.52	402.33	1:264
2000	2.28	575.52	1:252

Table 1: Comparison of the recursive and non-recursive algorithms on the VAX 11/780 computer.

rithms are used to generate exactly the same images on a 80 × 80 lattice and the computation time required is used as a criterion for comparison. Table 1 shows the results. It is evident that the recursive algorithm is about two orders of magnitude more efficient than the non-recursive one.

5. Conclusion

In this paper, we described both non-recursive and recursive techniques for the identification of closed loops in diffusion-limited aggregation processes. The elimination of closed loops from DLA images can greatly reduce the computational effort required in generating these images. The recursive algorithm developed can identify closed loops by checking relationship among a few neighborhood points and utilize previous knowledge of the image at each stage of the aggregation process. Scanning the complete image at each stage is not necessary. It was shown, by simulation, that the recursive algorithm is a couple of orders more efficient than the non-recursive one.

Acknowledgements

We would like to thank Dr. Harvey Scher for bringing our attention to the closed loop identification problem in DLA images and for his review of this paper. We also want to thank Lee Ann Fitzer for her assistance in FORTRAN programming and David Stickford for his assistance in LISP programming.

References

[1] S. K. Friedlander, *Smoke, Dust, and Haze*, (Wiley, New York, 1977).

[2] W. W. Mullins and R. F. Sekerka, *Journal of Applied Physics*, **34** (1963).

[3] T. A. Witten and L. M. Sander, "Diffusion-limited Aggregation," *Physical Review B*, **27(9)** (1983).

[4] T. A. Witten and L. M. Sander, "Diffusion-limited Aggregation, a Kinetic Critical Phenomenon," *Physical Review Letters*, **47** (1981).

[5] A. Rosenfield and A. C. Kak, *Digital Picture Processing*, (Academic Press, 1982).

[6] L. A. Turkevich and H. Scher, "Occupancy-Probability Scaling in Diffusion-limited Aggregation," *Physical Review Letters*, **55** (1985).