Percolation and cluster structure parameters: The enhanced Hoshen-Kopelman algorithm

J. Hoshen*
812 Shiloh Circle, Naperville, Illinois 60540-7112

M. W. Berry† and K. S. Minser‡
Department of Computer Science, University of Tennessee, Knoxville, Tennessee 37996-1301

(Received 12 March 1997)

An enhanced Hoshen-Kopelman (EHK) algorithm for calculating cluster structure parameters is presented for the site percolation problem. The EHK algorithm enables efficient calculation of cluster spatial moments, perimeters, and bounding boxes for very large Monte Carlo simulated lattices. The algorithm is used to compute the squared radius of gyration and internal perimeters of clusters in 3000×3000 square simulated lattices. The squared radius of gyration was used to calculate the value of the correlation length exponent. [S1063-651X(97)01408-6]

PACS number(s): 64.60.−i

I. INTRODUCTION

In 1976, Hoshen and Kopelman introduced in a Physical Review B paper [1], to which we refer as I, a breakthrough algorithm for cluster analysis in percolation phenomena. Today, this algorithm is known as the Hoshen-Kopelman (HK) algorithm [2] or the Hoshen-Kopelman method [3]. Before the publication of the HK algorithm, typically lattices containing several tens of thousands of sites were analyzed. Using the HK algorithm, in 1984, Margolina *et al.* [4] analyzed lattices containing approximately 1.7×10^{10} sites. Rapaport [5] in 1986 raised that number to 2.56×10^{10} sites. The existing world record, established in 1992 by Rapaport [6], is reported for a lattice containing 4.096×10^{11} sites.

The success of the HK algorithm can be attributed to its linear time and superior space computational complexities [7] in terms of lattice size. Because the HK algorithm requires only a single and sequential pass through the lattice, it allows for a very efficient utilization of computer memory hierarchies [e.g., random access memory (RAM), cache] where only a small fraction of the lattice is actually stored in memory (or cache) at any point in time. Given recent progress in parallel implementations [8–11] of the HK algorithm, it is conceivable that the algorithm could be applied to lattices having over 10^{14} sites (which is still not quite Avogadro's number).

While initially the HK algorithm primary use was in the domain of pure and basic sciences, later it began percolating into applications in diverse fields of technology and applied sciences. Some examples of such applications are in nuclear fuel rod processing [12], catalysis [13], curing of epoxy resin [14], classification of radar signatures [15], and assessing habitat fragmentation [16]. Furthermore, the HK algorithm is being used in health-related areas such as evaluating bone structures in osteoporosis [17]. In contrast, the algorithm has also been used in analyzing the somewhat *health questionable* preparation of deep-fat-fried tortilla chips [18].

The HK algorithm has been very successful in determining cluster sizes for very large lattices. Nevertheless, it does not provide information on cluster structure. In Sec. II we shall describe the enhanced Hoshen-Kopelman (EHK) algorithm that will enable us to determine cluster shape parameters while preserving the time and space complexities of the original HK algorithm. In Sec. III we shall illustrate how a shape parameter such as the squared radius of gyration is calculated using the EHK algorithm. Correlation length results that are based on the radius of gyration computation as well as perimeter calculations will be presented in Sec. IV. Finally, some limitations of the EHK algorithm will be discussed in Sec. V.

II. THE ENHANCED HK ALGORITHM

The algorithm described here is a natural enhancement of the original HK algorithm discussed in I. The original algorithm classifies by size clusters of A type molecules in a binary lattice containing randomly distributed mixture of A and B molecules. We assume that the concentration of A molecules is p, which is also the probability that a lattice site is occupied by A.

The HK algorithm assigns a cluster label m_t^α to each lattice site occupied by A where α is a cluster identifier. This algorithm allows multiple label assignment to a cluster α. The labels are a set of natural numbers:

$$\{m_1^\alpha, m_2^\alpha, \ldots, m_s^\alpha, \ldots, m_t^\alpha, \ldots\}. \tag{1}$$

At least one cluster site is labeled with one of the labels given by Eq. (1). In this set one number is considered the *proper* cluster label. We choose the smallest number, m_s^α, to be the *proper* label. The choice of the smallest cluster number as the *proper* label is not mandatory. Choosing any other label in Eq. (1) would be appropriate. The following set of integers defines the relationship between the cluster labels:

$$\{N(m_1^\alpha), N(m_2^\alpha), \ldots, N(m_s^\alpha), \ldots, N(m_t^\alpha), \ldots\}. \tag{2}$$

*Electronic address: jhoshen@att.net.
†Electronic address: berry@cs.utk.edu
‡Electronic address: minser@cs.utk.edu

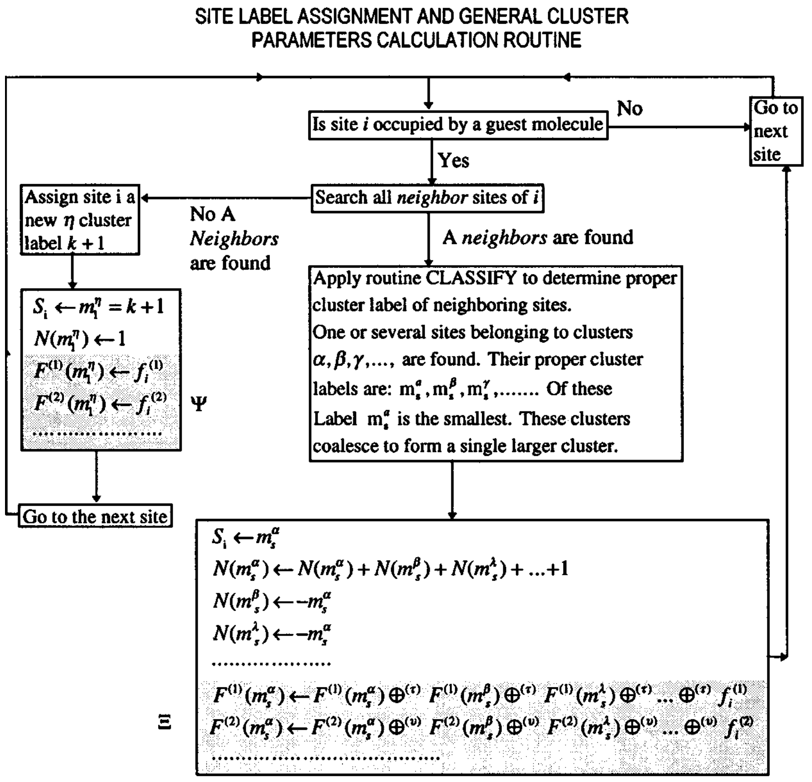
SITE LABEL ASSIGNMENT AND GENERAL CLUSTER PARAMETERS CALCULATION ROUTINE
Is site i occupied by a guest molecule
No
Go to next site
Yes
Search all *neighbor* sites of i
No A *Neighbors* are found
Assign site i a new η cluster label $k+1$
A *neighbors* are found
Apply routine CLASSIFY to determine proper cluster label of neighboring sites. One or several sites belonging to clusters $\alpha, \beta, \gamma, \ldots$, are found. Their proper cluster labels are: $m_s^\alpha, m_s^\beta, m_s^\gamma, \ldots$ Of these Label m_s^α is the smallest. These clusters coalesce to form a single larger cluster.
$S_i \leftarrow m_1^\eta = k+1$
$N(m_1^\eta) \leftarrow 1$
$F^{(1)}(m_1^\eta) \leftarrow f_i^{(1)}$
$F^{(2)}(m_1^\eta) \leftarrow f_i^{(2)}$
Ψ
Go to the next site
$S_i \leftarrow m_s^\alpha$
$N(m_s^\alpha) \leftarrow N(m_s^\alpha) + N(m_s^\beta) + N(m_s^\lambda) + \ldots + 1$
$N(m_s^\beta) \leftarrow -m_s^\alpha$
$N(m_s^\lambda) \leftarrow -m_s^\alpha$
$F^{(1)}(m_s^\alpha) \leftarrow F^{(1)}(m_s^\alpha) \oplus^{(\tau)} F^{(1)}(m_s^\beta) \oplus^{(\tau)} F^{(1)}(m_s^\lambda) \oplus^{(\tau)} \ldots \oplus^{(\tau)} f_i^{(1)}$
$F^{(2)}(m_s^\alpha) \leftarrow F^{(2)}(m_s^\alpha) \oplus^{(\upsilon)} F^{(2)}(m_s^\beta) \oplus^{(\upsilon)} F^{(2)}(m_s^\lambda) \oplus^{(\upsilon)} \ldots \oplus^{(\upsilon)} f_i^{(2)}$
Ξ

FIG. 1. Site label assignment routine of cluster labels in the EHK algorithm for molecules of type A.

In Eq. (2), $N(m_s^\alpha)$ is the only positive integer member of the set. It denotes the number of A molecules belonging to the cluster. The rest of the numbers in Eq. (2) are negative integers. They link the labels m_t^α with the *proper* label m_s^α. These labels are related to m_s^α via the following set of equations:

$$m_r^\alpha = -N(m_t^\alpha), m_q^\alpha = -N(m_r^\alpha), \ldots, m_s^\alpha = -N(m_t^\alpha). \quad (3)$$

Equation (3) represents a tree graph, where the root of the tree corresponds to the *proper* label m_s^α. All the other labels are nodes on the tree pointing directly, or indirectly through other nodes, to the root.

The multiple labeling technique is used in conjunction with the Monte Carlo method where a pseudorandom number generator is used to create the random lattice. Sites are created sequentially. Lattice sites are generated column by column (or row by row). Once a column is filled, the next column starts filling. In three-dimensional lattices, layers of sites are created in sequential order. As sites are created, they are assigned cluster labels as shown in Fig. 1 and described in I. At each site, previously labeled neighbor sites are inspected. Routine CLASSIFY of Fig. 2 determines the *proper* labels of these neighboring sites. The result is that only a single scan is required to create the lattice and determine the cluster size distribution.

The enhancements to the HK algorithm are presented by the gray shaded areas in Fig. 1. The shaded area, Ψ, denotes initialization of some general cluster properties $F^{(1)}(m_s^\alpha)$, $F^{(2)}(m_s^\alpha), \ldots$ by quantities $f^{(1)}(i)$, $f^{(2)}(i), \ldots$, respectively. The $f^{(n)}(i)$ quantities represent some properties of ith lattice site. For example, $f^{(1)}(i)$ could indicate that site i is on the cluster boundary. It would be zero if all neighbor sites are of type A, and 1 if at least one of its neighbors is of type B. Another $f^{(n)}(i)$ could denote the X coordinate, x_i, of the ith site. $f^{(n)}(i)$ could also be a nonscalar quantity. For example, $f^{(n)}(i) = (x_i^2, y_i^2, z_i^2)$.

The operator denoted by $\oplus^{(\tau)}$ in the shaded area, Ξ, defines a general binary HK operator. The only requirement on this operator is that it would be associative and commutative. Usually it would denote some kind of addition operation. But

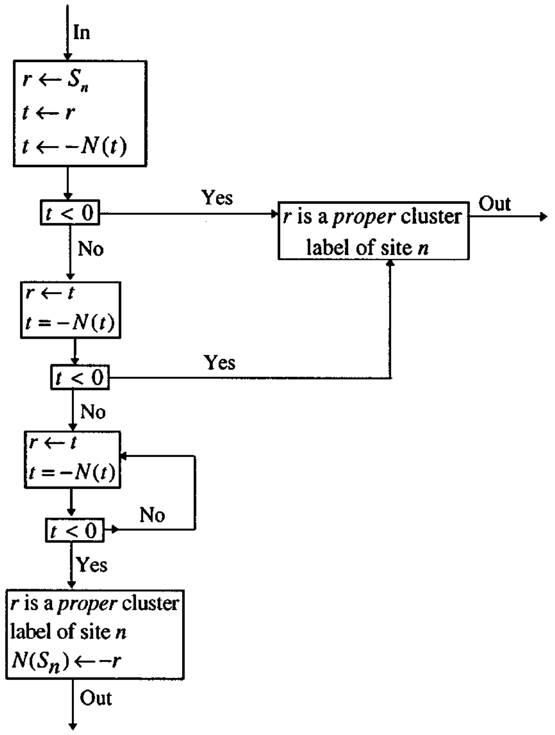

FIG. 2. Routine CLASSIFY for determining *proper* labels of neighboring A sites.

other operations that meet these requirements could also be used. For example, the functions max(a,b) and min(a,b) can be used to define a HK operator such as $x_i \oplus^{(\kappa)} x_j \equiv \max(x_i, x_j)$, and $x_i \oplus^{(\chi)} x_j \equiv \min(x_i, x_j)$. These operators could be used to determine bounding rectangles and boxes for clusters in two dimensions and three dimensions, respectively. The $F^{(n)}$ quantities in Fig. 1 are calculated from cluster fragments α, β, γ that are merged into a single cluster. This is a generalization of the cluster size calculation in the original HK algorithm where the size of the merged cluster α is the sum of the cluster fragments $\alpha, \beta, \gamma, \ldots$.

Spatial moments are very important quantities in image analysis [19]. They can be calculated using the EHK algorithm, where the HK operator is defined as the scalar addition operator. These moments are defined as

$$X_j^{(n)}(k) = \sum_{i \in k} x_j^n(i). \tag{4}$$

$X_j^{(n)}(k)$ denotes the nth spatial moment of the kth cluster for the jth coordinate, where $j = 1, 2, \ldots, d$ and d is the dimension of the lattice. The summation in Eq. (4) is carried over all sites i of cluster k. The original HK algorithm can be viewed as a special case of Eq. (4) when $n = 0$.

A major feature of the original HK algorithm was that columns (or layers) could be divided into sections where cluster labels would be reused. This reduces the number of labels required for a very large lattice. When clusters extend from one section to the next section, they are assigned a new *proper* label that belongs to the set of labels of the next section. Just as labels are recycled, storage space for the $F^{(n)}$ quantities can also be recycled. The implication is that these generalized quantities could be calculated for essentially the same size of lattices analyzed by the original HK algorithm.

III. THE SQUARED RADIUS OF GYRATION AND CORRELATION LENGTH

The squared radius of gyration, R_s^2, is an important cluster shape quantity [2]. It is defined by

$$R_s^2 = \frac{1}{2s^2} \sum_i \sum_j |\mathbf{r}_i - \mathbf{r}_j|^2, \tag{5}$$

where s is the size of the cluster, and $\mathbf{r}_i$ and $\mathbf{r}_j$ denote the positions of sites i and j, respectively. The double summation involves all i and j sites of the cluster of interest. The squared distances in Eq. (5) can be rewritten as

$$|\mathbf{r}_i - \mathbf{r}_j|^2 = \sum_{\mu=1}^{d} (x_{i,\mu} - x_{j,\mu})^2, \tag{6}$$

where $x_{i,\mu}$ and $x_{j,\mu}$ denote the μth coordinates of sites i and j, respectively, in d dimensions.

Since each dimension μ can be treated independently in Eq. (6), the summation in Eq. (5) can be reduced for each μ to

$$\sum_i \sum_j (x_{i,\mu} - x_{j,\mu})^2 = 2s \sum_i x_{i,\mu}^2 - 2\left(\sum_i x_{i,\mu}\right)^2 = 2[sX_\mu^{(2)} - (X_\mu^{(1)})^2], \tag{7}$$

where $X^{(1)}$ and $X^{(2)}$ are the first and second moments, respectively, for the given cluster and μth coordinate. Using Eqs. (5)–(7) we get

$$R_s^2 = \frac{1}{s^2} \sum_{\mu=1}^{d} [sX_\mu^{(2)} - (X_\mu^{(1)})^2]. \tag{8}$$

Notice that R_s^2 in Eq. (8) can be computed in terms of the first and second cluster spatial moments, which can be determined by the EHK algorithm.

Following Stauffer and Aharony [2], using R_s^2, the squared correlation length [20], ξ^2, can be computed by

$$\xi^2 = \frac{2\sum_s s^2 \sum_{t=1}^{n_s} R_{st}^2}{\sum_s s^2 n_s}, \tag{9}$$

where n_s denotes the number of clusters of size s. R_{st}^2 specifies the squared radius of gyration of a cluster animal t of size s.

The EHK algorithm facilitates the simultaneous calculation of n_s and R_{st}^2 during lattice traversal. Whenever a column in a two-dimensional lattice or a plane is completed in a three-dimensional lattice, all clusters that do not extend through the column or plane are *completed* clusters. Hence, as clusters are enumerated the partial sums of s^2 and $s^2 R_{st}^2$ can be accumulated. When lattice scanning is completed, using Eq. (9), the ratio of these sums yields ξ^2. The primary advantage of this approach is that individual R_{st}^2 values need not be stored in computer memory.

IV. NUMERICAL EXAMPLES

Computations of cluster parameters using the EHK algorithm are given in Table I. These data are given for Monte Carlo runs for 3000×3000 size lattices and for p values in the range of 0.2–0.61 for a square lattice. For each data point in the table, a new pseudorandom number seed is used. The table gives information on cluster sizes and internal cluster perimeters for the largest cluster. In addition to that, the squared radius of gyration for the largest cluster, $R_s^2(\max_s)$, and the squared radius of gyration for the cluster with the maximum squared radius of gyration, $\max_{R_s^2}$, is also given. From this table it is quite evident that the largest cluster does not necessarily have the largest squared radius of gyration. This is not surprising because the squared radius of gyration is related to the average distance between two cluster sites. This average distance depends on the specific shapes of the clusters in question.

The correlation length ξ in Table I is calculated from the squared correlation length given by Eq. (9). The correlation length exponent ν is determined by plotting $\ln(p_c - p)$ versus $\ln(\xi)$, as shown in Fig. 3. The value of p_c was taken to be

TABLE I. Cluster statistics for square lattices of size 3000×3000. In the table, p denotes the site occupation probability, n denotes the number of clusters in the lattice, $\max_s$ denotes the size of the largest cluster by the number of sites occupied, $T(\max_s)$ denotes the ratio of the perimeter of the largest cluster to the largest cluster size, $R_s^2(\max_s)$ denotes the squared radius of gyration for the largest cluster, $\max_{R_s^2}$ denotes the cluster with the largest squared radius of gyration, ξ denotes the correlation length, and t_{sec} denotes the elapsed CPU time in seconds for each sample lattice run on a Sun Ultra 2 workstation.

p	n	$\max_s$	$T(\max_s)$	$R_s^2(\max_s)$	$\max_{R_s^2}$	ξ	t_{sec}
0.200	1094061	27	0.926	2.842	3.307	1.415	5.265
0.300	1154703	77	0.987	6.849	6.849	2.402	6.519
0.400	955518	165	0.945	7.637	9.643	4.584	7.211
0.500	592669	901	0.900	21.117	27.327	12.275	7.316
0.510	552527	911	0.913	24.019	24.019	14.191	7.256
0.520	511781	1115	0.893	22.461	26.939	17.024	7.235
0.530	472869	2070	0.878	36.366	38.532	21.245	7.221
0.540	434099	2167	0.892	36.855	38.984	25.107	7.221
0.550	395152	3373	0.894	53.367	53.367	32.598	7.195
0.555	376813	5278	0.877	50.551	67.785	41.633	7.203
0.560	358236	6814	0.874	69.388	75.152	48.200	7.197
0.565	339097	7664	0.887	88.308	88.308	59.670	7.162
0.570	322208	18301	0.877	124.480	124.480	78.942	7.194
0.575	305227	22215	0.876	137.730	137.730	112.916	7.178
0.580	289980	50061	0.872	239.800	239.800	171.613	7.212
0.585	274178	79545	0.868	244.360	244.360	236.833	7.209
0.590	257374	426408	0.864	473.160	510.720	629.269	7.226
0.593	248745	1305977	0.861	908.010	908.010	1187.097	7.215
0.600	229206	4001264	0.857	1200.400	1200.400	1697.498	7.435
0.610	204472	4631412	0.850	1221.100	1221.100	1726.934	7.480

0.592 746 [2]. The values of p used for this calculation range from $p=0.5$ to $p=0.58$. This p range was chosen because it had the least deviation from linearity in the ln-ln plot. Points above 0.58 would lead to a significant deterioration in the linear fit. Such behavior for Monte Carlo calculations of exponents near p_c was previously reported by Hoshen *et al.* [21]. The value of ν for Fig. 3 is 1.331 ± 0.012, which is in good agreement with ν values determined by Levinshtein *et al.* [22], Kapitulnik *et al.* [23], and Hoshen *et al.* [21].

The ratio of the number of internal perimeter sites to the cluster size for the largest cluster in the sample is displayed in Fig. 4. It initially rises but then falls as p_c is approached.

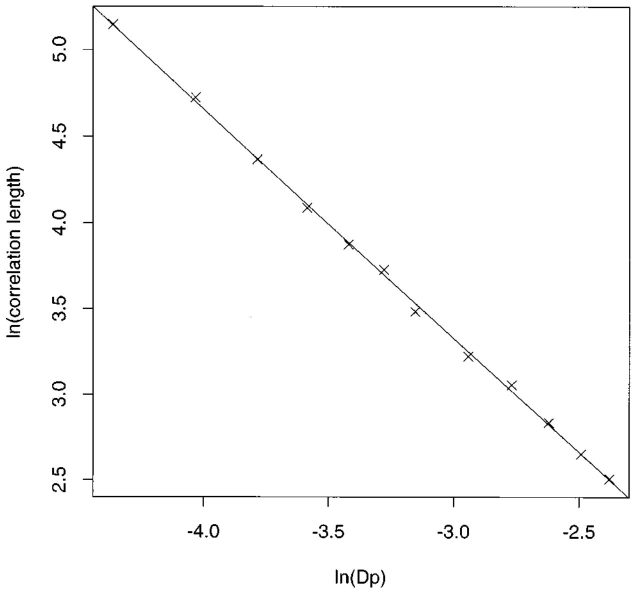

FIG. 3. Natural logarithm ln of the correlation length ξ versus ln(Dp) where $Dp=p_c-p$. p_c is the critical percolation probability. The data points denoted by × are taken from Table I. The solid line is a linear regression line for these data points.

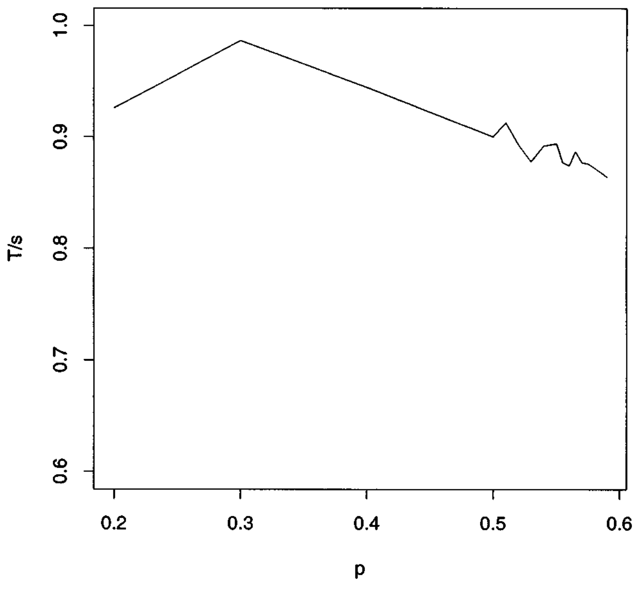

FIG. 4. The ratio of T, cluster perimeter, to s, cluster size, for the largest cluster in the sample as a function of p.

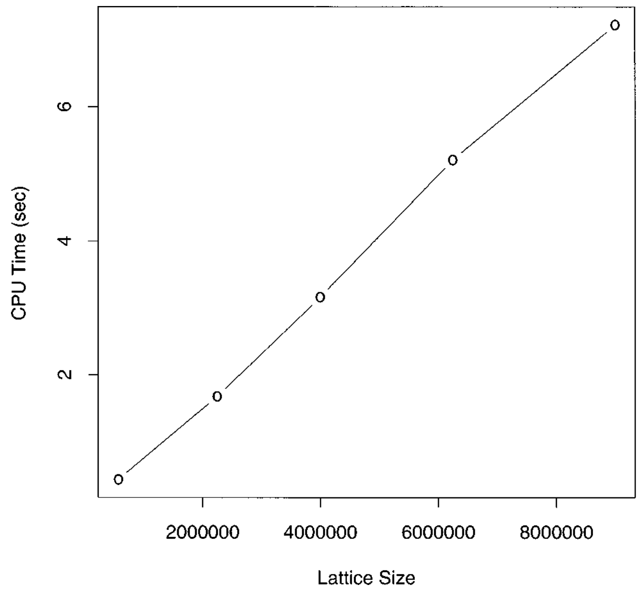

FIG. 5. CPU time in seconds for sample lattices denoted by ○ for sizes 750×750, 1500×1500, 2000×2000, 2500×2500, and 3000×3000.

The fluctuations between $p=0.5$ and $p=0.6$ could be related to fluctuations that are normally seen near p_c. Below $p=0.5$ the ratio appears to be smoother, but this could be related to the fact that this area is represented by fewer data points.

The last figure, Fig. 5, illustrates the linear behavior of the computer central processing unit (CPU) computation time as a function of the lattice size. Clearly, the computational time complexity of the original HK algorithm is preserved by the EHK algorithm.

V. DISCUSSION

The computations given in this paper demonstrate the potential of applying the EHK algorithm for very large lattices for calculation of cluster shape parameters. In our example, we used the squared radius of gyration to calculate the correlation length exponent ν. Instead, we could have calculated ν by computing the bounding squares of the clusters [23]. The EHK algorithm can determine the clusters' bounding squares using the min and max operators as described in Sec. II. Such calculation would require only a single pass over the lattice.

With all its potential, the EHK algorithm has some limitations. In percolation calculations, the external cluster perimeter has usually been the parameter of interest [24]. Yet, we used the EHK algorithm to calculate the internal cluster perimeter. The reason for this choice is that there is no simple way to calculate the external cluster perimeter using the EHK algorithm. The problem is that when two cluster fragments are merged (see Fig. 1), their perimeters would also be merged. The intersection of the internal perimeter sites is the null set. Therefore it is possible to add up their counts in the cluster fragment merger. In contrast, the intersection of the external perimeter sites may not produce the null set. The implication is that a simple addition of the counts of the external sites is not possible. So, to use the EHK algorithm for the external perimeter sites, the HK operator would have to correspond to the *union* operation. The union operator meets the EHK algorithm requirements for associativity and commutativity. Unfortunately, using the union operator would require keeping lists of perimeter sites for each cluster fragment, and merging these lists when clusters are merged. Using such lists will significantly increase the run time and memory space utilization for the EHK algorithm.

Despite some limitation, the EHK algorithm is clearly a major step forward in cluster analysis. Probably, its most important use would be in calculating cluster spatial moments. The computation of these moments, under the EHK algorithm, is very efficient and is likely to provide new information on the structural properties of clusters.

ACKNOWLEDGMENT

M.W.B. and K.S.M. were supported in part by The National Science Foundation under Grant No. NSF-ASC 94-11394 and No. NSF-CDA 95-29459.

[1] J. Hoshen and R. Kopelman, Phys. Rev. B **14**, 3438 (1976).
[2] D. Stauffer and A. Aharony, *Introduction to Percolation Theory*, revised 2nd ed. (Taylor & Francis Ltd., London, 1994), p. 168.
[3] *Fractals and Disordered Systems*, revised 2nd ed., edited by A. Bunde and S. Havlin (Springer-Verlag, Berlin, 1996), p. 97.
[4] A. Margolina, H. Nakanishi, D. Stauffer, and H. E. Stanley, J. Phys. A **17**, 1683 (1984).
[5] D. C. Rapaport, J. Phys. A **19**, 291 (1986).
[6] D. C. Rapaport, J. Stat. Phys. **66**, 679 (1992).
[7] The time and memory space needed by an algorithm expressed as a function of the size of the problem, which is in our case the size of the lattice, see, for example, A. V. Aho, J. E. Hopcroft, and J. D. Ullman, *The Design and Analysis of Computer Algorithms* (Addison-Wesley, Reading, MA, 1974), p. 2.
[8] A. N. Burkitt and D. W. Heerman, Comput. Phys. Commun. **54**, 201 (1989).
[9] J. M. Constantin, M. W. Berry, and B. T. Vander Zanden, Int. J. Supercomputer Applications High Performance Computing **11(1)**, 34 (1997).
[10] H. Nakanishi, V. Rego, and V. Sunderam, J. Parallel Distrib. Comput. **24**, 177 (1995).
[11] M. Flanigan and P. Tamayo, Physica A **215**, 461 (1995).
[12] H. C. Kim and G. S. Cho, Ann. Nucl. Energy **23**, 1445 (1996).
[13] L. Zhang and N. A. Seaton, Chem. Eng. Sci. **51**, 3257 (1996).
[14] W. H. Jo and M. B. Ko, Macromolecules **27**, 7815 (1994).
[15] S. Haimov, O. I. Yordanov, M. A. Michalev, and A. Savchenko, IEEE Trans. Geosci. Remote Sens. **27**, 606 (1989).
[16] M. Berry, J. Comiskey, and K. Minser, IEEE Comput. Sci. Eng. **1**, 24 (1994).

[17] J. H. Kinney, N. E. Lane, and D. L. Haupt, J. Bone Mineral Res. **10**, 264 (1995).
[18] R. G. Moreira and M. A. Barrufet, J. Food Eng. **27**, 279 (1996).
[19] W. K. Pratt, *Digital Image Processing*, 2nd ed. (John Wiley, New York, 1991), p. 630.
[20] Equation (9) has a somewhat different form for the squared correlation length from the one given by Eq. (47b) of Ref. [2], p. 60. However, there is no difference between the two representations because the squared radius of gyration value in Eq. (47b) is an average value over all the animals, which is what is stated by Eq. (9).
[21] J. Hoshen, R. Kopelman, and J. S. Newhouse, J. Phys. Chem. **91**, 219 (1987).
[22] M. E. Levinshtein, B. I. Shklovskii, M. S. Sur, and A. L. Efros, Zh. Éksp. Teor. Fiz. **69**, 386 (1975) [Sov. Phys. JETP **42**, 197 (1976)].
[23] A. Kapitulnik, A. Aharony, G. Deutscher, and D. Stauffer, J. Phys. A **16**, L269 (1983).
[24] H. Franke, Z. Phys. B **40**, 61 (1980).