NODE DISTRIBUTION IN A PR QUADTREE*

Chuan-Heng Ang
Hanan Samet

Computer Science Department
Institute of Advance Computer Studies and
Center for Automation Research
University of Maryland
College Park, MD 20742

ABSTRACT

A method, termed *approximate splitting*, is proposed to model the node distribution that results when the PR quadtree is used to store point data drawn from a uniform distribution. This method can account for the aging and phasing phenomena which are common in most hierarchical data structures. Approximate splitting is also shown to be capable of being adapted to model the node distribution of the PR quadtree with points drawn from a known non-uniform distribution.

Keywords and phrases: PR quadtrees, population analysis, hierarchical data structures.

* The support of the National Science Foundation under Grant IRI-88-02457 is gratefully acknowledged.

1. INTRODUCTION

Geographic information systems store large amount of data including maps of roads and regions, locations of cities, etc. Each of these objects can be described with different data structures. For example, we may store the gray levels of a region map in an array, the locations of cities in lexicographical order of their coordinates, and the roads as a set of line segments which are determined by their end points. Ideally, these objects should be stored using variants of the same underlying data structure so that the effort used to implement and maintain the data structure can be kept to the minimum. This goal has been achieved in the implementation of **QUILT**, a geographic information system [Same84b]. In this system, the underlying data structure is the linear quadtree [Garg82].

The quadtree is a hierarchical data structure used to organize an object space. If the object space is an image plane (e.g., a region map), then the quadtree describing it is called a *region quadtree* as defined by Klinger [Klin71, Same84a]. The region quadtree decomposes an image into homogeneous blocks. If the image is all one color, then it is represented by a single block. If not, then the image is decomposed into quadrants, subquadrants, ..., until each block is homogeneous. If the object space is a set of line segments, then the quadtree that is used is called a *PM quadtree* [Same84a]. If the object space consists of points within a square, e.g., the locations of all the cities in certain region, then the corresponding quadtree is a *PR quadtree* [Same84a]. In this case the square is decomposed into quadrants, subquadrants, ... similarly until the number of points in each block is within a certain limit. This limit is termed the *node capacity* of the PR quadtree.

The quadtree variants that we described enable us to have a unified representation of three different types of objects encountered in a geographic information system, namely regions, points, and lines. The storage requirements of the region quadtree are analyzed in [Dyer82, Shaf88] and those of the PM quadtree are analyzed in [Same85]. In this paper, we focus on the storage requirements of the PR quadtree. Given n points which are to be stored in a PR quadtree, we show how to compute the storage requirements, the node distribution, and the average node occupancy. By learning more about the storage requirements of the PR quadtree, we will be able to predict the storage used by the PR quadtrees in a dynamic environment.

Nelson and Samet [Nels86] use a technique termed *population analysis* to analyze the node distribution in a PR quadtree. A *population* is defined to be the collection of all the nodes of specific occupancy. For example, all empty nodes form one population, nodes containing one point a second, and so forth. A node containing i points is said to be of type n_i. Adding a point to a node containing i points will either convert it to a node with $i+1$ points or cause the node to split and produce four nodes one level deeper with occupancies that vary between 0 and $m+1$ where m is the node capacity. This transformation is described by the corresponding *transformation matrix*. The fraction obtained by dividing the number of nodes in population i by the

total number of nodes in the PR quadtree is called the *population frequency* e_i. The vector $\vec{e}$ formed by all the e_i is called the *population frequency distribution.*

Population analysis assumes that the distribution of node occupancies is independent of the geometric size of the corresponding block. In addition, suppose that a steady state can be reached when the points are inserted dynamically, then a set of equations involving the population frequencies can be derived and solved by numerical method. The population frequency distributions calculated using this technique agree fairly well with the experimental results.

Given a population frequency distribution $\vec{e}$, the predicted average node occupancy can be computed by the dot product of $\vec{e}$ and the vector $(0,1,...,m)$. This number is determined solely by the node capacity and is not affected by the size of the PR quadtree. On the other hand, the actual average node occupancies obtained from our experiments show a cyclical variation which is periodic in the logarithm of the total number of points stored in PR quadtree. This is termed the *phasing* phenomenon by Nelson and Samet [Nels86] and it is shown in Figure 1. In addition, the PR quadtree also exhibits what is termed the *aging* phenomenon by Nelson and Samet [Nels86]. In this case, after a node is created, it is filled to its capacity as more and more points are inserted into it. In other words, the node is aging, or getting older. Our experiments also show that bigger blocks fill up faster. This is consistent with an analysis based on geometric probability [Sant76].

The population analysis method only provides us with the population frequency distribution. It does not indicate how many nodes containing i points are at depth j, i.e., the complete *node distribution.* Thus, we can not predict the average node access cost when the PR quadtree is stored in main memory. The population analysis ignores the aging phenomenon. Also, the average node occupancy derived from the population frequency distribution is fixed regardless of the size of the PR quadtree. Therefore, the population analysis falls short in accounting for the phasing phenomenon.

Since the population analysis method fails to reflect the phasing and the aging phenomena, the two important characteristics of the PR quadtree, we must look for an alternative analysis technique. In this paper, we propose a method termed *approximate splitting* to calculate the approximate values of the average node distribution. This enable us to derive an approximation of the population frequency distribution. We will refer to these approximations as the *predicted node distribution* and the *predicted population frequency distribution,* respectively. For a PR quadtree that is built from a set of random points, i.e., points that are generated from a uniform distribution, we can obtain its *actual node distribution* and *actual population frequency distribution.* In order to ease the comparison with [Nels86], we will use the same notation and examples that they used.

The remainder of this paper is organized as follows. Section 2 introduces the approximate splitting method through an example. Section 3 describes the method. Section 4 compares the results obtained using the approximate splitting method with those in [Nels86]. Section 5 generalizes the method so that it can be applied to the PR quadtree with data points drawn from a non-uniform distribution such as a Gaussian distribution. Section 6 discusses the aging and phasing phenomena. Section 7 discusses the discrepancy between the predicted average node occupancy obtained by using the approximate splitting method and the actual average node occupancy. We draw conclusions in Section 8.

2. THE AVERAGE NODE DISTRIBUTION

Let us first consider a PR quadtree with data points drawn independently from a uniform distribution. The probability that a point will fall within a particular region is proportional to the area of that region. Given a region that has been partitioned into four quadrants, the probability that a point falls within a particular quadrant is $\frac{1}{4}$, and the probability that it falls outside the quadrant is $\frac{3}{4}$. Thus the probability for a node that contains two points to have an empty NW quadrant is $\binom{2}{0}(\frac{3}{4})^2$. Since there are four quadrants in a node, the average number of empty quadrants is $4\times\binom{2}{0}(\frac{3}{4})^2=\frac{9}{4}$. Similarly, the average number of quadrants with one point is $4\times\binom{2}{1}(\frac{1}{4})(\frac{3}{4})=\frac{3}{2}$ and the average number of quadrants with two points is $4\times\binom{2}{2}(\frac{1}{4})^2=\frac{1}{4}$.

When a PR quadtree contains only one point, it has only one node and its average node distribution is trivial. When more than one point is being stored in a quadtree that has a node capacity of one, we have to consider the splitting process. Figure 2 shows an example of a PR quadtree with node capacity one that contains two points. Its actual node distribution is 8 empty nodes and 2 full nodes. In general, a PR quadtree with node capacity m and maximum depth n may contain t points that are drawn from a uniform distribution. We adopt the convention that the root node N is at level n (or depth 0) and a pixel-size node is at level 0 (or depth n). In order to understand the problem involved in calculating and approximating the node distribution, let us look at the following example.

Example 1: Find the average node distribution of the PR quadtrees containing 1000 points with node capacity one and of maximum depth 9.

We can compute by brute force the predicted average node distribution for Example 1 as follows. Since the root node contains 1000 points, it is split to produce some nodes at depth 1. In particular, at depth 1, the expected number of empty nodes is $4\times\binom{1000}{0}(\frac{3}{4})^{1000}$, the expected number of nodes containing 1 point is $4\times\binom{1000}{1}(\frac{1}{4})(\frac{3}{4})^{999}$, the expected number of nodes containing 2 points is $4\times\binom{1000}{2}(\frac{1}{4})^{2}(\frac{3}{4})^{998}$, ..., and the expected number of nodes containing 1000 points is $4\times\binom{1000}{1000}(\frac{1}{4})^{1000}$. All nodes at depth 1 containing more than one point will be split to produce more nodes at depth 2. To calculate the number of empty nodes at depth 2, we need to find out the number of nodes produced from the splitting of nodes at depth 1 of types $n_2, n_3, ..., n_{1000}$. These numbers are summed up to give the number of nodes at depth 2 of type n_0. Similarly, we have to find out the number of nodes of types n_1, ..., n_{1000} at depth 2. This calculation is carried out up to depth 9 which is the maximum resolution. The average node distribution as well as the average population distribution can then be found.

The amount of calculation involved is so huge that it makes the calculation of the average node distribution by this brute force method impractical. It can be used only when t is small. When t is large, we are content if we can find a good approximation of the average node distribution. In the next section, we describe a method which produces an approximate average node distribution.

3. THE APPROXIMATE SPLITTING PROCESS

Suppose that we already know how to calculate the average node distribution for a subtree, say S, of a certain height. For simplicity, assume that the PR quadtree consists of multiple copies of S. Each subtree's corresponding blocks are of the same size and contain the same number of points. Since the points are uniformly distributed over blocks of the same size, each subtree will have the same average node distribution. Using this assumption, we can now approximate the average node distribution of the splitting process originating at the root node of the PR quadtree by combining the average node distributions of all these subtrees. We use the term *approximate splitting* to describe this method. The required calculation can be divided into three steps.

Step 1 : Find the depth s such that the probability of having a leaf node at depth d, $d \leq s$, is smaller than a predetermined small value. In other words, we want to determine the depth at which we are quite sure that all nodes will require splitting.

Since a node will be split when it contains more than m points, we first find the largest integer s such that $f=\frac{t}{4^s}>m$. That is, we distribute the t points into all the nodes at depth s so that each node contains f points. This value of s may not be the right choice. For instance, in Example 1, since $\frac{1000}{4^4}=\frac{1000}{256}>1$ and $\frac{1000}{4^5}<1$, we have $s=4$.

Next, we want to estimate the average number of leaf nodes at depth s that can possibly be produced during the splitting process. In particular, we want to find the number of full nodes thus produced. At depth $s-1$, there are 4^{s-1} nodes and each of them contains $4f$ points. A node at depth $s-1$ can be split to produce a node with m points at depth s with probability $p_s=\binom{4f}{m}(\frac{1}{4})^m(\frac{3}{4})^{4f-m}$. In other words, it can split and produce $4p_s$ full nodes. There are 4^{s-1} nodes at depth $s-1$ and hence there are $4^s\times p_s$ full nodes which are leaf nodes at depth s. If this number is smaller than a small constant ε, say 0.1, then we know that when the splitting process begins at depth $s-1$, the number of nodes at depth s containing m points will be so small that these full leaf nodes can be ignored. For a binomial expansion $(p+q)^n$ with $p+q=1$ and $p<q$, all the terms in the expansion before the i^{th} term where $i=np$ are monotonically increasing. When $n=4f$ and $p=\frac{1}{4}$, $m<f=4f\times\frac{1}{4}=np$. Thus the probability of splitting a node at depth $s-1$ with $4f$ points to produce a node at depth s with j points, $j<m$, is even smaller than that for producing the nodes with m points. We can safely say that splitting a node at depth $s-1$ will not produce any leaf nodes at depth s (i.e., nodes with m or fewer points). In other words, the probability that all the nodes at depth s are GRAY nodes is very high. Thus we can stop moving up the tree and prepare to split the nodes at depth s.

If $4^s\times p_s\geq\varepsilon$, then we can set s to $s-1$, and f to $4f$, and repeat the test to see whether it is necessary to move up the tree again.

For Example 1, $m=1$, $f=\frac{1000}{4^4}>1$, $s=4$, $4f\approx 16$ and $p_s\approx\binom{16}{1}(\frac{1}{4})(\frac{3}{4})^{15}=0.008634$. Therefore $4^4\times p_s=3.4>0.1$. This means that we have to move up one level and now we have $f\approx 16$, $s=3$, $p_s\approx\binom{64}{1}(\frac{1}{4})(\frac{3}{4})^{63}<0.0000001$ and $4^3\times p_s<0.1$. Therefore we may decide to split the nodes at depth 3 with $k=4^3=64$ subtrees each containing $f=\frac{t}{k}=\frac{1000}{64}=15.625$ points.

Suppose that we would have split the nodes at depth 4 instead of at depth 3. In this case, the nodes produced by this splitting process are at depth 5 or deeper and we

have unnecessarily forced the points to be distributed over too many subtrees. On the other hand, if we split the nodes at depth 2 instead of depth 3, we will produce a more accurate result at the expense of more calculation (i.e., the brute force algorithm). This is shown in the empirical results that are tabulated in the next section. In general, to produce a result which is more accurate, we should try to split the nodes as high up the tree as possible as long as this does not cause any computational problems such as arithmetic overflow or underflow, or running out of memory.

Step 2 : Calculation of the initial node distribution.

After we have determined the value of s, we know that all the nodes will appear at depth $s+1$ and deeper. The initial node distribution of the nodes at depth $s+1$ will be $k \times \binom{t/k}{j} \frac{3^{t/k-j}}{4^{t/k-1}}$ where $0 \le j \le t/k$ and $k=4^s$.

Step 3 : Calculate the approximate average node distribution.

After we have obtained the initial node distribution for those nodes at depth $s+1$, we can repeat the splitting process for those nodes with more than m points until the maximum depth of the tree is reached, at which depth all nodes with more than m points will be treated as nodes with only m points. The approximate average node distribution thus obtained is the *predicted node distribution* .

4. COMPARISON OF THE RESULTS

In this section, we compare the results predicted by using the approximate splitting method and those predicted by Nelson and Samet's method [Nels86]. For comparison, we also show the node distributions and the population frequency distributions obtained from PR quadtrees built by the insertion of 1000 randomly generated points. Table 1 shows the actual and predicted node distribution for Example 1. The predicted node distribution fits very well with the actual node distribution.

Table 1. Node distribution of PR quadtree containing 1000 points.				
Depth	Actual		Approximate Splitting Method	
	n_0	n_1	n_0	n_1
0	0.0	0.0	0.0	0.0
1	0.0	0.0	0.0	0.0
2	0.0	0.0	0.0	0.0
3	0.0	0.0	0.0	0.0
4	6.6	20.1	4.7	19.4
5	300.2	354.2	309.6	361.5
6	533.7	411.6	536.4	405.6
7	225.4	144.9	226.1	155.3
8	71.5	49.6	64.5	43.3
9	16.1	19.5	16.7	14.9

In our comparison, we consider PR quadtrees with node capacity ranging from 1 to 8, and each PR quadtree contains 1000 random points. From the predicted node distribution, we can obtain the predicted population frequency distribution by dividing the number of nodes of each population by the total number of nodes in the quadtree. In Table 2, we show the population frequency distribution $\vec{e_i}$ obtained using the different analyses. $\vec{e_1}$ is obtained using the method in [Nels86], $\vec{e_2}$ is obtained by analyzing the set of 64 subtrees at depth 3 with $\frac{1000}{64}$ points apiece, $\vec{e_3}$ is obtained by analyzing the set of 16 subtrees at depth 2 with $\frac{1000}{16}$ points apiece, and $\vec{e_4}$ is obtained from the PR quadtrees which are built through insertion of 1000 random points. $\vec{e_1}$ and $\vec{e_4}$ are taken from [Nels86].

Table 2. Average population frequency distributions of PR quadtrees.

---

m=1: $\vec{e_1}$=(.500,.500)

$\vec{e_2}$=(.538,.462)

$\vec{e_3}$=(.538,.462)

$\vec{e_4}$=(.536,.464)

m=2: $\vec{e_1}$=(.278,.418,.304)

$\vec{e_2}$=(.332,.426,.242)

$\vec{e_3}$=(.330,.425,.245)

$\vec{e_4}$=(.326,.427,.247)

m=3: $\vec{e_1}$=(.165,.320,.305,.210)

$\vec{e_2}$=(.221,.359,.268,.151)

$\vec{e_3}$=(.216,.358,.273,.153)

$\vec{e_4}$=(.213,.364,.273,.149)

m=4: $\vec{e_1}$=(.102,.239,.276,.225,.158)

$\vec{e_2}$=(.141,.279,.263,.187,.130)

$\vec{e_3}$=(.140,.282,.270,.186,.122)

$\vec{e_4}$=(.139,.293,.264,.184,.120)

m=5: $\vec{e_1}$=(.065,.179,.238,.220,.172,.126)

$\vec{e_2}$=(.078,.190,.232,.211,.170,.119)

$\vec{e_3}$=(.083,.204,.244,.208,.155,.106)

$\vec{e_4}$=(.084,.217,.241,.204,.151,.104)

m=6: $\vec{e_1}$=(.043,.132,.200,.207,.176,.137,.105)

$\vec{e_2}$=(.037,.117,.190,.221,.202,.148,.086)

$\vec{e_3}$=(.046,.139,.208,.216,.181,.130,.080)

$\vec{e_4}$=(.050,.150,.201,.215,.176,.127,.081)

m=7: $\vec{e_1}$=(.028,.098,.165,.189,.173,.143,.114,.090)

$\vec{e_2}$=(.019,.077,.160,.219,.219,.166,.097,.044)

$\vec{e_3}$=(.027,.099,.177,.214,.197,.146,.091,.049)

$\vec{e_4}$=(.034,.110,.177,.214,.187,.143,.091,.044)

m=8: $\vec{e_1}$=(.019,.073,.135,.168,.166,.145,.119,.097,.078)

$\vec{e_2}$=(.013,.063,.147,.216,.224,.172,.102,.047,.017)

$\vec{e_3}$=(.019,.079,.160,.209,.203,.155,.097,.052,.025)

$\vec{e_4}$=(.024,.086,.151,.206,.194,.156,.100,.049,.034)

From the above data, we see that splitting the nodes nearer the root produces better results, i.e., $\vec{e_3}$ is more accurate than $\vec{e_2}$. We also notice that $\vec{e_3}$ matches $\vec{e_4}$ quite closely. Table 3 shows the average node occupancies AVG_i for the distributions shown in Table 2. The *average node occupancy* can be found by computing the dot product of $\vec{e_i}$ with the vector $(0,1,2,...,m)$.

The last column AVG_4/m shows the average storage utilization of the PR quadtree with node capacity m. The average of all the entries in this column is 0.474. It is an approximation of the average storage utilization of all the PR quadtrees of node capacities ranging from 1 to 8.

Table 3. Average node occupancy.

Node capacity (m)	AVG_1	AVG_2	AVG_3	AVG_4	AVG_4/m
1	0.50	0.46	0.46	0.46	0.460
2	1.03	0.91	0.92	0.92	0.460
3	1.56	1.35	1.36	1.36	0.453
4	2.10	1.89	1.87	1.85	0.463
5	2.63	2.56	2.47	2.44	0.488
6	3.17	3.22	3.06	3.03	0.505
7	3.72	3.65	3.50	3.44	0.491
8	4.25	3.83	3.76	3.79	0.475

5. APPLICATIONS OF APPROXIMATE SPLITTING

Suppose that we are given a PR quadtree with data points drawn from a known non-uniform distribution. If we can partition the quadtree into subtrees such that each of them can be regarded as a PR quadtree with points drawn from a certain uniform distribution, then the complicated splitting process of the original PR quadtree can be modeled by the combination of the splitting processes of all the subtrees in the partition. With such an adaptation, the approximate splitting process can have a wider application. In the following, we will look at two examples.

Example 2: Consider an application where the locations of the houses in a certain area are captured in a PR quadtree. As shown in Figure 2, it is known that the housing density in a particular sector, say A which is 1/16 of the area, is double the housing density in the rest of the area. Can we predict the node distribution if we know that there are 1000 houses in the area given that at most one house is associated with each node?

To solve this problem, we divide the area into 16 squares and regard each of them as a subtree rooted at a quadtree block at depth 2. Sector A contains $\frac{2}{17}\times 1000$ points and each of the other 15 subtrees has $\frac{1}{17}\times 1000$ points. The predicted node distribution of the PR quadtree is the sum of the predicted node distribution of sector A and 15 times the predicted node distribution of any other sector. The result is shown in Table 4. This result matches very well with the actual node distribution that is obtained by generating three PR quadtrees and taking the average of their node distributions.

Table 4. Node distribution of the PR quadtree in example 2.

Depth	Actual		Predicted	
	n_0	n_1	n_0	n_1
0	0.0	0.0	0.0	0.0
1	0.0	0.0	0.0	0.0
2	0.0	0.0	0.0	0.0
3	0.0	0.0	0.0	0.0
4	5.00	28.33	5.38	21.12
5	296.33	334.66	303.62	350.92
6	522.66	402.33	526.49	401.18
7	246.66	182.00	229.75	158.31
8	57.66	40.00	66.44	44.66
9	12.66	12.66	17.23	13.45

Example 3: Suppose that a PR quadtree has 1000 points in a 512 by 512 square and that the x and y coordinates are independently distributed according to the Gaussian distribution with mean 256 and standard deviation 128.

For this example, the square contains the points that lie within two standard deviations from the mean. Since the points will cluster around the mean value, we would like to divide the square into smaller areas around the center of the square as shown in Figure 3. This partition also reflects the fact that the leaf nodes corresponding to the blocks surrounding the center appear at deeper levels of the PR quadtree. Let a=0.3830 be the probability for the x (or y) coordinate of a point to fall within 0.5 standard deviation from the mean and let b=0.6826 be that for 1 standard deviation. The probability that a point which is generated according to this specific Gaussian distribution will fall into a particular region can be calculated as follows. There are 4 subtrees of type A with total probability $(\frac{a}{2})^2 \times 4$; 8 subtrees of type B with probability $(\frac{b-a}{2})\times\frac{a}{2}\times 8$; 4 subtrees of type C with probability $(\frac{b-a}{2})^2 \times 4$; 8 subtrees of type D with probability $(\frac{1-b}{2})\times\frac{b}{2}\times 8$; and 4 subtrees of type E with probability $(\frac{1-b}{2})^2\times 4$. The number of points contained in each subtree is just the product of the probability associated with the subtree and 1000. The node distribution of each subtree can then be calculated. The sum of the node distributions of all the subtrees is the predicted node distribution of the PR quadtree containing 1000 points. Table 5 shows that the predicted node distribution fit well with the actual node distribution that is obtained by taking the average of the node distributions of three PR quadtrees built from 1000 random points generated according to a Gaussian distribution.

Table 5. Node distribution of a PR quadtree with points drawn from a Gaussian distribution.

Depth	Actual		Predicted	
	n_0	n_1	n_0	n_1
0	0.00	0.00	0.00	0.00
1	0.00	0.00	0.00	0.00
2	0.00	0.00	0.00	0.00
3	0.33	0.00	0.01	0.09
4	28.00	40.00	16.15	35.34
5	236.00	257.33	259.13	296.41
6	502.66	417.00	511.13	406.96
7	291.00	192.33	266.34	185.92
8	98.00	77.00	81.46	54.93
9	18.00	18.00	21.44	16.75

6. AGING AND PHASING

Nelson and Samet's analysis [Nels86] computes the average population frequency distribution for a PR quadtree with a certain node capacity. As long as the node capacity remains the same, no matter how many points the PR quadtree holds, the average population frequency distribution of the PR quadtree remains the same, and so is the predicted average node occupancy. This is obviously not true. Therefore there is a discrepancy between the actual average node occupancy and the average node occupancy predicted by their method. They attribute the discrepancy to two factors termed phasing and aging that could not be explained by population analysis.

On the other hand, phasing can be demonstrated by the approximate splitting method. In Figure 4, the average node occupancies of the PR quadtrees with node capacity 8 as predicted by the approximate splitting method are plotted against the number of points stored in the PR quadtrees. The actual average node occupancies are superimposed in the same graph for easy comparison. Figure 4 shows clearly that the graph does oscillate and is periodic in the logarithm of the total number of data items stored in the trees.

Aging is responsible for the fact that larger nodes have occupancies in excess of the predicted average occupancy of the whole PR quadtree. This is accounted for in the approximate splitting process as can be seen from Table 6 which shows the average node occupancy by node level for Example 1. The predicted node distribution for Example 1 is shown in Table 1 and its average node occupancy is 0.46 which is the first entry of the column with heading AVG_4 in Table 3. Comparing with the number 0.46, we find that larger nodes in the predicted node distribution do have higher occupancies. The data in Table 6 is obtained from Table 1 as follows. For nodes at depth 4, the actual node occupancy is $\frac{20.1}{6.6+20.1}=0.75$ and the predicted average node occupancy is $\frac{19.4}{4.7+19.4}=0.81$. Similarly, we can compute the rest of the ratios in the table. From the table, we see that the predicted average node occupancy is smaller for smaller nodes. The anomalously high value for the actual average node occupancy at depth 9 is the result of the implementation which truncates the tree at that depth [Nels86].

Table 6. Average node occupancy by node level.		
Depth	Actual	Predicted
0	0.00	0.00
1	0.00	0.00
2	0.00	0.00
3	0.00	0.00
4	0.75	0.81
5	0.54	0.54
6	0.44	0.43
7	0.39	0.41
8	0.41	0.40
9	0.55	0.47

7. DISCREPANCY

In Figure 4 we see that although the curve of the predicted average node occupancy oscillates in step with that of the actual average node occupancy, the predicted values achieve higher peaks and lower valleys. This is the result of the assumption that all subtrees partitioning a PR quadtree contain the same number of points. According to this assumption, when a subtree achieves its maximum of the average node occupancy, so have all other subtrees when the approximate splitting method is used. If this constraint is relaxed by allowing the subtrees to have different number of points as in the real situation, then the values of their average node occupancies will spread so that the average of these values will have less variation and hence the oscillation of the curve is dampened.

The number of points falling into the root nodes of the subtrees at depth s follows the binomial distribution with parameters t and p, where t is the number of points contained in the PR quadtree, and p is the probability that a random point falls into the root node of a particular subtree of the approximate splitting process. For a PR quadtree of maximum depth n, we have $p=4^{-(n-s)}$. When t is large, the binomial distribution can be approximated by a normal distribution with mean $\mu=tp$ and standard deviation $\sigma=\sqrt{tp(1-p)}$ [Fell57, page 172]. The probability that the number of points in a subtree is within one standard deviation from the mean value is about 0.68. Since we are only interested in the approximation of the average node occupancy, one simple improvement that can be made to reduce the amplitude of the curve of the predicted average node occupancy is to divide the subtrees into three groups containing $\mu-\sigma$ points, μ points, and $\mu+\sigma$ points in the proportion of 20, 60, and 20. This proportion is chosen because about 60 percent of the points fall within one standard deviation from the mean. Figure 5 shows the curves of the actual and modified predicted (labeled Predicted1) average node occupancies. The corresponding data can be found in Table 7. It is clear that the accuracy of the predicted average node occupancy has been improved noticeably after the modification.

Table 7. Variation in the average node occupancy.

Number of points	Actual	Modified predicted	Predicted
250	3.846	3.773	3.763
300	4.052	4.147	4.343
350	4.204	4.258	4.494
400	3.977	3.951	4.315
500	3.671	3.585	3.668
600	3.426	3.387	3.280
700	3.282	3.364	3.201
800	3.392	3.432	3.317
900	3.558	3.604	3.530
1000	3.762	3.783	3.763
1100	3.904	3.946	4.127
1200	4.081	4.088	4.343
1300	4.146	4.158	4.466
1400	4.154	4.177	4.494
1500	4.135	4.168	4.433
1600	4.025	3.950	4.315
1700	3.919	3.812	4.155
1800	3.825	3.759	3.986
1900	3.708	3.680	3.819
2000	3.618	3.588	3.668
2100	3.540	3.553	3.537
2200	3.462	3.478	3.429
2300	3.427	3.467	3.343
2400	3.380	3.413	3.280
2500	3.355	3.364	3.235
2600	3.322	3.374	3.209
2700	3.332	3.343	3.198
2800	3.334	3.393	3.201
2900	3.349	3.380	3.217
3000	3.366	3.411	3.242
3100	3.397	3.415	3.276
3200	3.428	3.446	3.317
3300	3.462	3.493	3.365
3400	3.488	3.519	3.417
3500	3.537	3.561	3.472
3600	3.587	3.597	3.530
3700	3.630	3.653	3.588
3800	3.674	3.694	3.647
3900	3.714	3.732	3.706
4000	3.754	3.778	3.763

8. CONCLUDING REMARKS

The approximate splitting process is a means to obtain an approximation of the node distribution and the average population frequency distribution of a PR quadtree. It can account for the aging and phasing phenomena and gives a fairly accurate prediction of both distributions. It does not resort to solving the equations using numerical methods as that was done in [Nels86]. The approximate splitting process can also be adapted to PR quadtrees with the points drawn from a known non-uniform distribution. It is useful in estimating the storage requirements as well as the performance of the PR quadtree when it is built in main memory.

The process of partitioning a given PR quadtree with the points drawn from a known non-uniform distribution is similar to the way we construct a step function to approximate an arbitrary function. Since the partitioning greatly depends on the prior knowledge of the non-uniform distribution, it is difficult to design a scheme to automate the partitioning process of any given PR quadtree, although such a scheme is desirable.

9. REFERENCES

[Dyer82] - C.R. Dyer, The space efficiency of quadtrees, *Computer Graphics and Image Processing 19*, 4(August 1982), 335-348.

[Fell57] - W. Feller, *An Introduction to Probability Theory and its Applications*, Volume 1, second edition, John Wiley, New York, 1957.

[Garg82] - I. Gargantini, An effective way to represent quadtrees, *Communications of the ACM 25*, 12(December 1982), 905-910.

[Klin71] - A. Klinger, Patterns and Search Statistics, in *Optimizing Methods in Statistics*, J.S. Rustagi, Ed., Academic Press, New York, 1971, 303-337.

[Nels86] - R.C. Nelson and H. Samet, A population analysis of quadtrees with variable node size, Computer Science TR-1740, University of Maryland, College Park, MD, December 1986.

[Same84a] - H. Samet, The quadtree and related hierarchical data structures, *ACM Computing Surveys 16*, 2(June 1984), 187-260.

[Same84b] - H. Samet, A. Rosenfeld, C.A. Shaffer, and R.E. Webber, A geographic information system using quadtrees, *Pattern Recognition 17*, 6(1984), 647-656.

[Same85] - H. Samet and R.E. Webber, Storing a collection of polygons using quadtrees, *ACM Transactions on Graphics 4*, 3(July 1985), 182-222.

[Sant76] - L.A. Santalo, *Integral Geometry and Geometric Probability,,* Addison-Wesley, Reading, MA, 1976, Chapters 1-3.

[Shaf88] - C.A. Shaffer, A formula for computing the number of quadtree node fragments created by a shift, *Pattern recognition letters 7*, 1(January 1988), 45-49.

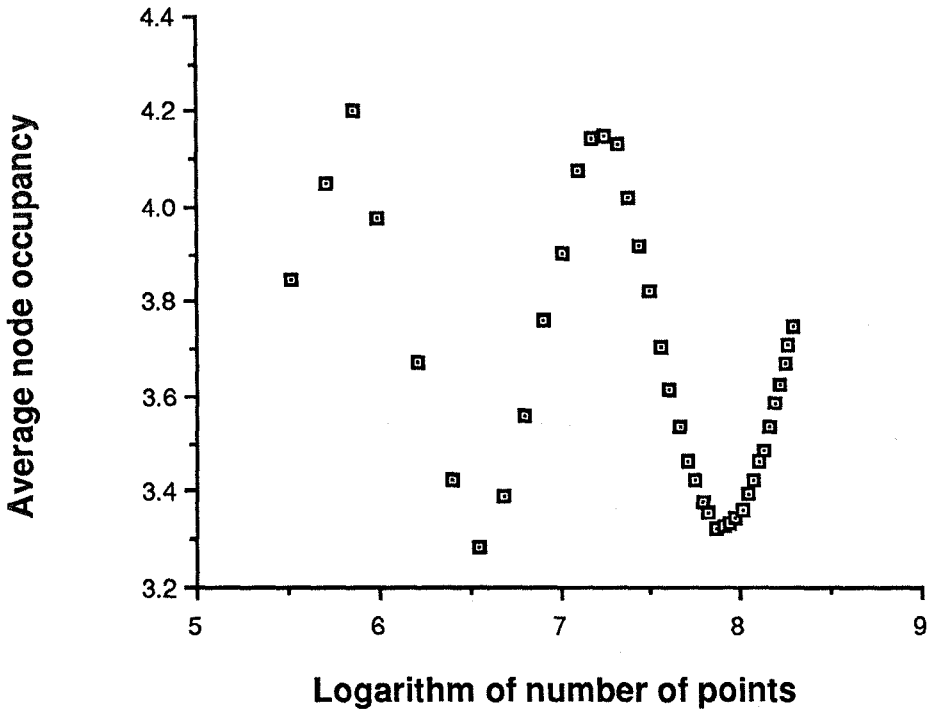

Figure 1 Average node occupancy of a PR quadtree.

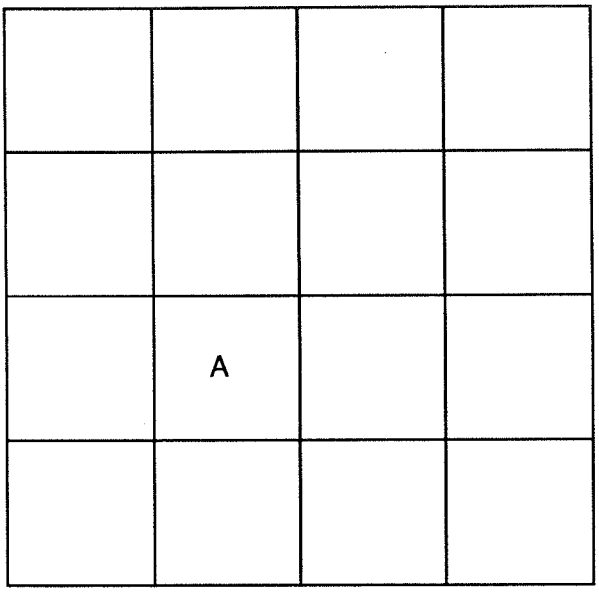

Figure 2 Partition of a residential area.

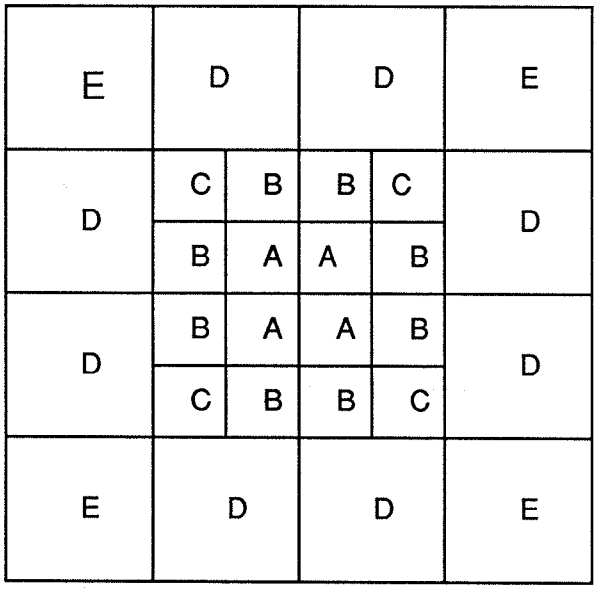

Figure 3 Partition of a square containing the points whose coordinates follow Gaussian distribution.

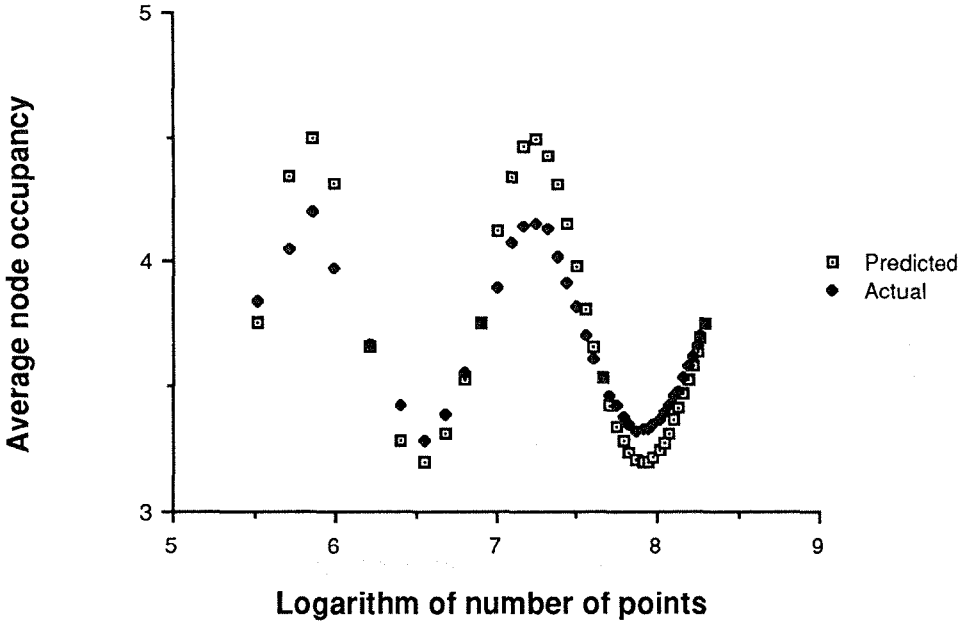

Figure 4 Comparison between the predicted and actual average node occupancies.

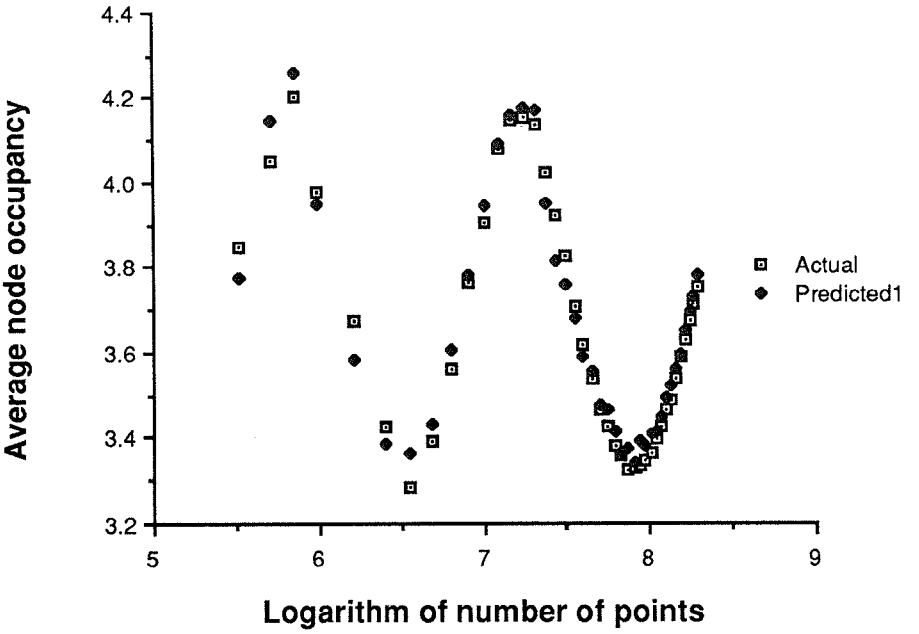

Figure 5 Comparison between the modified predicted and the actual average node occupancies.