QUADTREE REPRESENTATIONS OF DIGITAL TERRAIN

Zi-Tan Chen

Environmental Systems Research Institute
38Ø New York Ave. Redlands, CA 92373, USA.

Waldo R. Tobler

Geography Department
University of California
Santa Barbara, CA 931Ø6, USA

ABSTRACT

Digital elevations, or any other sampled scalar function (gray tones, densites) of two variables, can be approximated in several ways by a quadtree. The data domain is cut into four quadrans recursively and the elevation data within each quadrant are replaced by an approximating mathematical surface. The approximations evaluated are planes (average, maximum and minimum), a ruled surface, and a quadric surface. Tradeoffs between accuracy, computational speed, and storage cost suggest that the most appropriate approximation is the ruled surface.

INTRODUCTION

Most quadtree studies are only concered with binary images, whose pixels are either 'black' or 'white' (Samet 1984). It would be of value if this could be extended to cover other cases. In particular most geographic information systems attempt to store terrain as digital values at a lattice of locations. Continuous surfaces of this kind occur frequently in geographic problems and there is thus considerable value in attempting to represent terrain surfaces by quadtrees.

PREVIOUS WORK

A pointer structure was used to describe quadtrees in most early works, and no serious attention was given to the efficiency of manipulating large data sets. The pointer structure does not discriminate between binary or continuous images. For instance, such quadtrees have been conceptually used for image edge enhancement and image smoothing, that are fundamental digital image processing procedures (Ranade 1981). Quadrants through which edges pass, or that have a busy texture, will correspond to branch nodes in the tree, while those that are sufficiently homogeneous will be represented by leaf nodes. The meansure of homogeneity used for classifying a quadrant as a leaf or branch is important not only for the size of the tree but also for the quality of the image when it is reconstructed from the tree.

After quadtrees had been proposed for use in a geographic information system (GIS), efficiency became a consideration because one of important features of a GIS is its large volume of data and relations (Samet 1983, Tomlinson 1984, Chen 1985). An excellent storage-efficient method of representing a terrain surface by quadtrees has been proposed by Dutton (1983). His model considers the continuous propoty of terrain surface. In this scheme vertical and horizontal variation is related in that there is only small vertical variation with a small horizontal distance. All nodes of the quadtree nodes are relative to the elevation of the center point of its parent-node. One disadvantage of this scheme is that any operation involving a part of the image has to begin from the central point of the whole image.

Recently a "linear" method has been proposed as one efficient way to represent quadtrees. It has also been suggested for the representation of terrain surface. One converts the order of pixels from the row by row (raster) to the Morton number sequence. Thus the terrain surface is a compatible linear quadtree with those from binary images (Cebrian et al. 1985). This scheme stores all the lowest leaves of a quadtree but omits all nodes at higher levels of the tree. It loses the hierachical properties, which are important advantages of a quadtree structure. For a large flat area, this scheme and the raster image both use same number of pixels to represent the region. There is much redundant information.

What we expect for quadtrees representing a terrain is not only a compatibal data structure, but also a compact hierarchical data structure. The method introduced in this paper uses a linear quadtree to represent a terrain surface, as proposed in Cebrian's paper. The difference in this paper is in how to build the nodes at higher hierarchical levels of the terain quadtree. Thus this study is a preliminary study of efficient and practical methods to incorporate terrain data within a quadtree oriented GIS.

LINEAR QUADTREE REPRESENTATION

Using pointers to represent a quadtree is natural but is not efficient of storage. After converting a raster image into its quadtree, each node has an unique address name (its Morton number). The linear quadtree representation of a binary image only records the "black" nodes by noting their addresses. Each pixel is either filled (black) or blank (white), and thus it is only necessary to record where the "black" nodes exist.

The case becomes more complex when the image represents a terrain surface (or any continuously valued image). It is necessary not only to point out where the object is, but also to note its value because a node may be neither "black", nor "white". For continous surfaces, we must also record intermediate colors such as "dark grey", "light grey", "grey", etc.

Second, it is rare that there are many neighboring pixels within a large area which have an identical altitude. So building quadtree nodes at higher levels cannot depend on finding a large area with the exact same value. However, many mathematical equations can be used to approximate terrain surfaces. Based on this analogy, a hierarchy can be obtained. The values of the pixels within the quadtree node can be represented as a two-dimensional surface.

$$Z = F(i,j) \qquad where \;\; i_1 \leq i \leq i_2,\; j_1 \leq j \leq j_2 \tag{1}$$

To simplify calculations, the coordinates are normalized from 0.0 to 1.0. Figure 1 shows a terrain quadtree node and its four corners. The rows run from j1 to j2, while the columns are from i1 to i2. The actural terrain surface F(i,j) can be approximated by a mathematical function f(x,y) satisfing

$$\delta = |\; F(i,j) - f(x,y) \;| < \varepsilon \qquad where \quad 0 \leq x,y \leq 1. \tag{2}$$

Here ε is a given error range, and δ is an absolute difference. Using the absolute error instead of the more usual squared error criterion is somewhat arbitrary but is simple to implement in programs.

A linear quadtree of a terrain surface can be constructed in the following steps:

(1) Begin with the root node of a quadtree, set it as the current refrence quadtree node.

(2) Suppose a mathematical surface f(x,y) covers the current reference quadtree node. At its four corners, the values of the function f(x,y) are known from the values at the altitudes of the corners, respectively.

$$\begin{aligned} F(i_1 j_1) &= f\;(0,0) \\ F(i_2 j_1) &= f\;(1,0) \\ F(i_1 j_2) &= f\;(0,1) \\ F(i_2 j_2) &= f\;(1,1) \end{aligned} \tag{3}$$

We need to find the all parameters of the function f(x,y) from these corner values. A particular function f(x,y) can then be defined for this quadtree node.

(3) Check the difference δ at each of those pixels within the node. If δ at all pixels is less than the given error range ε, the current reference quadtree node can be replaced by the particular function f(x,y). Then go to step (5). If there is at least one pixel where δ is greater than the given error range ε, then this node can not be replaced by the function f(x,y). Go to step (4).

(4) Cut the current reference quadtree node into its four sub-quadrants. Put the four ndes into a queue of reprence quadtree nodes. Go to step (5).

(5) If there is at least one node in the queue, the first node of the queue of the reference quadtree nodes is set as the current reference quadtree node, and go to step (2). If there is not any node in the queue, go to step (6).

(6) Store each quadtree node which is well approximated by the particular mathematical surface. These nodes are stored in the order of the Morton number, which the parameters of the particular function f(x,y).

THE SURFACE FUNCTIONS f(x,y)

Five kinds of functions f(x,y) of surfaces are evaluated here. They all are tested using the previous procedure (especially step 2) to approximate a terrain surface, and have different CPU time and storage efficiencies

The average surface

For a normalized square (see Fig.1), the average height Za of a continuous surface can be calculated by:

$$Z_a = \int_{x=0}^{x=1}\int_{y=0}^{y=1} f(x,y)\, dx\, dy. \tag{4}$$

For a quadtree node, the average height represented in discrite form is:

$$Z_a = \sum_{i=i_1}^{i=i_2}\sum_{j=j_1}^{j=j_2} F(i,j) \,/\, (i_2-i_1)\,(j_2-j_1). \tag{5}$$

Whenever the difference δ is greater than the given error ε, it means that the terrain surface within the quadtree node is too rough to be approximated by the average plane under the given accuracy requirment.

When the terrain surface within a quadtree node can be replaced by its average for the given error e, the two numbers can uniquely represent this node. These two numbers are the

(address of the quadtree node) (average altitude)

A terrain surface can be represented as a set of such pairs of numbers. The first number of each pair is the address of the quadtree node represented by a Morton number, the second number of each pair is its average altitude.

The upper surface

Instead of the average we may use the maximum value in a quadtree node,

$$Z_t = MAX\ (F(i,j)) \qquad where\ \ i_1 \le i \le i_2,\ j_1 \le j \le j_2 \tag{6}$$

The other steps and the resulting representation are same as in the case of the average. Such a surface may be of use for applications involving aircraft.

The lower surface

We also can use the minimum altitude of the terrain in a quadtree node to approximate its surface.

$$Z_b = MIN\ (F(i,j)) \qquad where\ \ i_1 \le i \le i_2,\ j_1 \le j \le j_2 \tag{7}$$

The procedures and their result are similar with using the upper surface.

The ruled surface

The equation of a ruled surface (hyperbolic parabaloid) is:

$$f(x,y) = ax + bxy + cy + e \tag{8}$$

For a quadtree node, the ruled function in discrete form is:

$$Z_{ij} = a*(i-i_1) + b*(i-i_1)*(j-j_1) + c*(j-j_1) + e \tag{9}$$

A ruled surface is uniquely determinated by the four parameters: a, b, c, and e. Suppose a ruled surface f(x,y) covers a quadtree node, and fits at its four corners. We have

$$Z_1 = f(0,0) = e$$
$$Z_2 = f(1,0) = a + e$$
$$Z_3 = f(0,1) = c + e \qquad (10)$$
$$Z_4 = f(1,1) = a + b + c + e$$

Solving this set of equations, the parameters a, b, c, and e can be found as

$$a = Z_2 - Z_1$$
$$b = Z_4 + Z_1 - Z_2 - Z_3 \qquad (11)$$
$$c = Z_3 - Z_1$$
$$e = Z_1$$

When all of the elevations in a quadrant satisfy the error restriction, the elevation in the quadrant can be approximated by a ruled surface (through its four corners). Five numbers can represent this node.

(address of the quadtree node) (Z1) (Z2) (Z3) (Z4)

The first number is the address of the quadtree node, which is a Morton number. The other four numbers are altitudes at the four corners, respectively.

Obviously, for each quadtree node, the information used for a ruled surface is 5/2 of the amount needed for an average, maximum or minimum plane. However, the total number quadtree nodes needed by ruled function may much less than by one of the three planes, under an identical error criterion. Reconstruction of the estimated terrain will involve only bilinear interpolation for the ruled surfaces.

The quadric surface

A quadric function is:

$$f(x,y) = ax^2 + bxy + cy^2 + e \qquad (12)$$

For a quadtree node, the quadric function in discrete form is:

$$Z_{ij} = a*(i-i_1)^2 + b*(i-i_1)*(j-j_1) + c*(j-j_1)^2 + e \qquad (13)$$

If a quadric surface covers a quadtree node, and joins with terrain surface at the four corners, its parameters can be calculated as

$$\begin{aligned} a &= Z_2 - Z_1 \\ b &= Z_4 + Z_1 - Z_2 - Z_3 \\ c &= Z_3 - Z_1 \\ e &= Z_1 \end{aligned} \qquad (14)$$

When the surface in a quadtree node can be approximated by a particular quadric surface (through its four corners), a set of five numbers can represent this node.

(address of the quadtree node) (Z1) (Z2) (Z3) (Z4)

The first number is the node address in Morton number of the quadtree node. The other four numbers are the altitudes at the four corners, respectively.

ANALYSIS OF RESULTS

Experiments

The previous five kinds of surface functions have been tested for two sample areas with different topographic features. The first one is a relatively smooth area at Rapid City near the Black Hills, in South Dakota (West 103°16'-103°21'; North 44°12'-44°14.5'). The second is a mountainous area inside of the Balck Hills area (West 103°55'-104°00'; North 44°19.5'-44°22'). Both samples are digital elevation model in raster format, having 128*128 pixels with 50m horizontal interval. The original raster image have 16,384 pixels, and occupy 32,768 bytes. Various given accuracies are also tested on these test data. The results of tests are given in Table 1 and Table 2.

Storage space for terrain quadtrees

First, the smaller the given error tolerance e is, the more quadtree nodes exist. This is easily understood. Second, rough terrain needs more storage space for a correspondingly large number of quadtree nodes. Finally, the ruled surface is the best mathematical surface to approximate terrain of those tested surface functions. It always required the smallest storage space when other conditions were the same, compared with planar or quadric surfaces.

CPU time for generating terrain quadtrees

The smaller the given error tolerance e, the longer the CPU time needed for generating a terrain quadtree. Usually a rough terrain surface requires a longer CPU time than does a flat surface. The ruled surface always used less CPU time than the other kinds of surfaces, even less than the planar surfaces, which was somewhat unexpected.

CONCLUSIONS

Based on comparison of both storage space and CPU time for quadtree generation on terrain, the ruled surface is the best one among those five mathematical surface functions. We have only examined a very limited set of surface functions for only two test data. Much experimentation remains to be done. For examples, spline function and least square approximation have not been examined, and continuity conditions on the terrain, or its derivative, have not been imposed. Furthemore the applicability and implementation of these techniques to processing operations, such as map generation or relief shading, still require study. Detailed comparisons with the works of Cebrian et al (1985) and Dutton (1983) are also required. Nevertheless the analysis indicates the potential for quadtree structures as applied to terrain data.

REFERENCES

Cebrian J.A., Mower J.E., and Mark D.M. (1985) "Analysis and Display of Digital Elevation Models within a quadtree-based Geographic Information System", AUTO-CARTO 7 proceedings, Washington D.C., pp.55-64.

Chen Z.T. and Peuquet D. (1985) "Quadtree spatial spectra guide: A fast spatial heuristic search in a large geographic information system, AUTO-CARTO 7 Proceedings, Washington D.C., pp.75-82.

Dutton G. (1983) "Efficient Encoding of Gridded Surfaces", Spatial Algorithms for Processing Land Data with a Microcomputer, Lincoln Institute for Land Policy, Cambridge MA.

Samet H. (1984) "The Quadtree and Related Hierarchical data structures", ACM Computing Surveys, Vol.16, No.2, June, pp.187-260.

Ranade S. (1981) "Use of Quadtrees for Edge Enhancement", IEEE Trans. on System, Man and Cybernetics, Vol. SMC-11, No.5, May, pp.370-373.

Ranade S. and Shneier M. (1981) "Using Quadtrees to smooth images", IEEE Trans. on System, Man and Cybernetics, Vol.SMC-11, No.5, May, pp.373-376.

Tomlinson R.F. (1984) "Geographic Information Systems - A New Frontier", Proceedings of the International Symposium on Spatial Data Handling, Zurich, pp.3-5.

Table 1. Test results for the Rapid City area. Representing smooth terrain by a quadtree

surface type	error tolerance (ε)	CPU time	qdtr nodes	sum of numbers	sum of bytes	quadtree /raster
average	2m	84.5	14023	28046	56092	171.2%
average	6m	47.6	6733	13466	26932	82.2%
average	10m	34.1	4153	8306	16612	50.7%
maxflat	2m	86.5	14941	29882	59764	182.4%
maxflat	6m	65.3	10375	20750	41500	126.7%
maxflat	10m	45.0	6913	13826	27652	84.4%
minflat	2m	87.1	14941	29882	59764	182.4%
minflat	6m	62.8	10375	20750	41500	126.6%
minflat	10m	45.2	6913	13826	27652	84.4%
ruled	2m	50.1	3841	19205	38410	117.2%
ruled	6m	35.6	2419	12095	24190	73.8%
ruled	10m	28.5	1660	8300	16600	50.7%
ruled	100m	18.5	1200	6400	12000	40.7%
quadric	2m	51.4	4021	20105	40210	122.7%
quadric	6m	41.4	2851	14255	28510	87.0%
quadric	10m	31.1	1855	9275	18550	56.6%

Table 2. Test results for the Black Hills, mountainous area. Representing hilly terrain by a quadtree

surface type	error tolerance (ε)	CPU time	qdtr nodes	sum of numbers	sum of bytes	quadtree /raster
average	2m	91.2	15571	31142	62284	190.1%
average	6m	64.1	10246	20492	40984	125.1%
average	10m	46.3	6496	12992	25984	79.3%
maxflat	2m	90.7	16030	32060	64120	195.7%
maxflat	6m	81.9	13501	27002	54004	164.8%
maxflat	10m	64.9	10477	20954	41908	127.9%
minflat	2m	91.0	16030	32060	64120	195.7%
minflat	6m	77.0	13501	27002	54004	164.8%
minflat	10m	62.2	10477	20954	41908	127.9%
ruled	2m	48.3	4060	20300	40600	123.9%
ruled	6m	40.7	3253	16265	32530	99.3%
ruled	10m	34.9	2368	11840	23680	72.3%
ruled	100m	28.9	1900	8937	19000	53.3%
quadric	2m	51.1	4093	20465	40930	124.9%
quadric	6m	47.4	3616	18080	36160	110.4%
quadric	10m	39.4	2770	13850	27700	84.5%

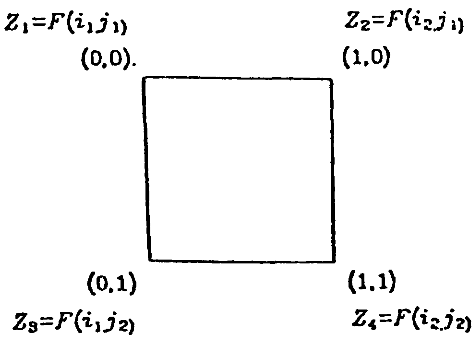

Figure 1. A normalized square