Title
Unit 36 - Hierarchical Data Structures

Authors
Unit 36, CC in GIS
National Center for Geographic Information and Analysis

Publication Date
1990

Peer reviewed

UNIT 36 - HIERARCHICAL DATA STRUCTURES

UNIT 36 - HIERARCHICAL DATA STRUCTURES

UNIT 36 - HIERARCHICAL DATA STRUCTURES

A. INTRODUCTION

- different scan orders produce only small differences in compression
  - the major reason for interest in Morton and other hierarchical scan orders is for faster data access
- the amount of information shown on a map varies enormously from area to area, depending on the local variability
  - it would make sense then to use rasters of different sizes depending on the density of information
    - large cells in smooth or unvarying areas, small cells in rugged or rapidly varying areas
  - unfortunately unequal-sized squares won't fit together ("tile the plane") except under unusual circumstances
    - one such circumstance is when small squares nest within large ones

there are, however, some methods for compressing raster data that do allow for varying information densities

B. INDEXING PIXELS

overhead - Raster to quadtree I and II

handout - Raster to quadtree

- consider the 16 by 16 array in which just one cell is different
  - notation: row and column numbering starts at 0
    - thus the odd cell is at row 4, column 7

Procedure

- begin by dividing the array into four 8x8 quadrants, and numbering them 0, 1, 2 and 3 as in the Morton order
  - quads 1, 2 and 3 are homogeneous (all A)
  - quad 0 is not homogeneous, so we divide only it into four 4x4 quads
    - these are numbered 00, 01, 02 and 03 because they are partitions of the 8x8 quad 0
    - of these, 00, 01 and 02 are homogeneous, but 03 is divided again into 030, 031, 032 and 033
    - now only 031 is not homogeneous, so it is divided again into 0310, 0311, 0312 and 0313

- what we have done is to recursively subdivide using a rule of 4 until either:

  - a square is homogeneous or
  - we reach the highest level of resolution (the pixel size)

- this allows for discretely adaptable resolution where each resolution step is fixed

- this concept is related to the use of Morton order for run encoding
  - if we had coded the raster using Morton order, each homogeneous square would have been a run
    - 8x8 squares are runs of 64 in Morton order, 4x4 are runs of 16, etc
  - the run encoded Morton order would have been:

    16A 16A 16A 4A 1A 1B 1A 1A 4A 4A 64A 64A 64A

  - if we allow runs to continue between blocks we could reduce this to:

    53A 1B 202A

  - i.e. a homogeneous block of 2m by 2m pixels is equivalent to a Morton run of 22m pixels

Decoding locations

- the conversion to row and column is the same as for decoding Morton numbers except that in this case the code is in base 4
  - in the example the lone B pixel is assigned code 0311

1. convert the code to base 2
   - hint: every base 4 digit converts to a pair of base 2 digits
   - thus 0311 becomes 00110101
2. separate the bits to get:
   - row 0100 = 4
   - column 0111 = 7

- so the numbering system is just the Morton numbering of blocks, expressed in base 4
- however, sequence and data compression are not the most useful aspects of this concept

C. THE QUADTREE

- can express this sequencing as a tree
  - the top is the entire array
  - at each level there is a four-way branching
  - each branch terminates at a homogeneous block
- the term quadtree is used because it is based on a rule of 4

overhead - Quadtree

- each of the terminal branches in the tree (the ones having values) is known as a leaf
  - in this case there are 13 leafs or homogeneous square blocks

Coding quadtrees

- to store this tree in memory, need to decide what to store in each memory location
  - there are many ways of storing quadtrees, but they all share the same basic ideas
- one way is to store in each memory location EITHER: 1. the value of the block (e.g. A or B), or or 2. a pointer to the first of the four "daughter" blocks at the next level down
- all four daughter blocks of any parent always occur together

overhead - Coding quadtrees

- thus, the quadtree might be stored in memory as:

Position: 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17

Contents: 2 6 A A A A A A 10 A 14 A A A B A A

(level):0 1 1 1 1 2 2 2 2 3 3 3 3 4 4 4 4

- the content of position 1 is a pointer indicating that the map is subdivided into four blocks whose contents can be found starting at position 2
- position 2 indicates that the four parts of the 0 block can be found beginning at position 6
- positions 3, 4 and 5 indicate that the other three level 1 blocks are all A and are not further subdivided

Accessing data through a quadtree

- consider two ways in which this quadtree may be accessed: 1. find all parts of the map with a given value 2. determine the contents of a given pixel

- notation: if the array has 2n by 2n pixels
  - there are n possible levels in the tree, or n+1 if we count the top level (level 0)
  - use m for the number of leafs

  1. to find the parts of a map with a given value we must examine every leaf to see if its value matches the one required
  - this requires m steps as there are m leafs

  2. to find the contents of a given pixel, start at the top of the tree
  - if the entire map is homogeneous, stop as the contents of the pixel are known already
  - if not, follow the branch containing the pixel
  - do know which branch to follow:
    - take the row and column numbers, write them in binary, interleave the bits, and convert to base 4

    - e.g. row 4, column 7 converts to 0311
    - at each level, use the appropriate digit to determine which branch to follow
    - e.g. for 0311, at level 0 follow branch 0, at level 1 follow branch 3, etc.
  - in the worst case, may have to go to level n to find the contents of the pixel, so the number of steps will be n

Comparison of different data structures

- summary of the work needed to perform the two types of queries:

  OptionFind parts with given valueFind contents of pixel

  Quadtree m n

  Row by row 22n (a) 1 (b)

  Row by row run encoded m (c) m (d)

  Morton run encoded m (c) m (d)

- notes: (a) must examine every pixel in the array (b) can calculate the position of every pixel and access it directly (c) the number of runs will be approximately the same as the number of leafs

    - although the orders are different, we decided earlier that order made little difference to the number of runs (d) each run must be examined to see if it contains the pixel

- thus, quadtree structures offer definite advantages over other systems for queries

D. VARIANTS OF QUADTREES

- the octree (or octtree) is a three-dimensional version of the quadtree, based on a rule of 8
  - the cube is divided recursively into eight pieces
  - octrees are useful for 3D data, particularly in mining and geology, and in medical imaging
  - see Unit 42 for more on this topic

- global data presents significant problems

  - we might use a projection such as the Mercator, and represent the data as a raster on this projection
  - the area and shape of the area represented by each pixel would be significantly distorted, particularly near the poles
  - the relationships between neighboring pixels would be distorted as well
    - in reality all of the pixels in the top and bottom rows are neighbors of each other across the poles

diagram

  - these problems create serious distortions in models based on such data

- a more suitable approach devised by Geoffrey Dutton is as follows: overhead/handout Global tesselation
  - project the globe onto an octahedron, consisting of eight triangles
    - the vertices of the octahedron are at the poles, and 90 degrees apart around the equator
  - number these from 0 through 7
  - each triangle is recursively divided using a rule of 4 into 4 smaller triangles, by connecting the midpoints of the edges
    - number these 0 (central triangle), 1 (vertically above or below), 2 (diagonally to the left) and 3 (diagonally to the right)
  - level 20 in this scheme has a resolution (triangle size) of about 1 m on the earth's surface
    - its address requires one base 8 digit and 20 base 4 digits, or 43 (binary) bits

E. ADVANTAGES OF HIERARCHICAL DATA STRUCTURES

- both coordinates are in a single address
  - a single number indicates a 2D location

- every square meter on the earth's surface has a consistent 21-digit address

- resolution is automatically known from the length of the address
    - in the global scheme described above, a 21-digit address has 1 m resolution, for 1 km resolution we need only a 13-digit address

- in comparison, a lat/long address needs two numbers, and it is not always easy to tell resolution from the way the numbers are presented

REFERENCES

Gargantini, I., 1982. "An effective way to represent Quadtrees," Communications of the ACM 25:905-910.

Mark, D.M., J.P. Lauzon and J.A. Cebrian, 1989. "A review of quad-tree based strategies for interfacing coverage data with digital models in grid form," International Journal of Geographical Information Systems 3(1):3-14.

Samet, Hanan, 1984. "The quadtree and related hierarchical data structures," ACM Computer Surveys 16(2):187-260.

Samet, Hanan, 1989. The Design and Analysis of Spatial Data Structures. Addison-Wesley, Reading, MA. Contains an excellent review of quadtrees.

Samet, Hanan, 1989. Applications of Spatial Data Structures. Addison-Wesley, Reading, MA. Contains an extensive review of applications of quadtrees.

Shaffer, C.A., H. Samet and R.C. Nelson, 1990. "QUILT: a geographic information system based on quadtrees," International Journal of Geographical Information Systems 4(2):103-32.

Waugh, Thomas C., 1986. "A response to recent papers and articles on the use of quadtrees for geographic information systems," Proceedings, Second IGU Symposium, Seattle, pp. 33-37. An interesting critique of quadtrees.

DISCUSSION AND EXAM QUESTIONS

1. "Hierarchical data structures are one of the few genuinely new concepts to come out of GIS research". Discuss.

2. Discuss the arguments presented for and against quadtrees in Waugh, 1986.

3 Summarize the arguments for and against the octahedral decomposition of the globe presented in the unit, as a means of representing and analyzing global databases.

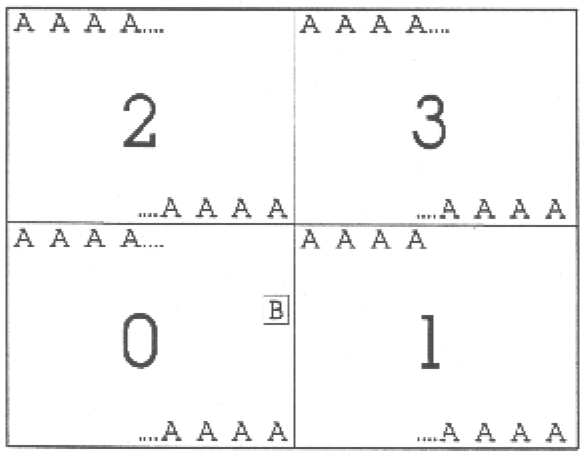
4. Modify the quadtree concept so that it is suitable as a method of storing digital elevation data. What are the advantages and disadvantages of this approach over a simple raster?

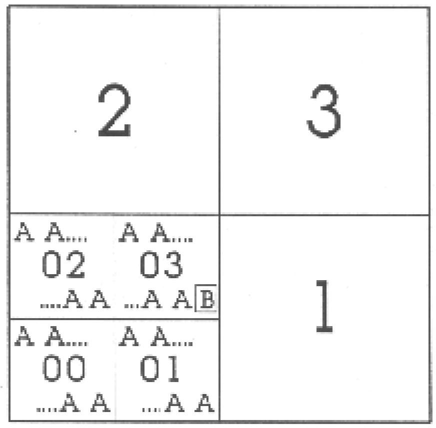
5. What would be the advantages and disadvantages of using Dutton's global tesselation as the primary means of discrete georeferencing, rather than street address?

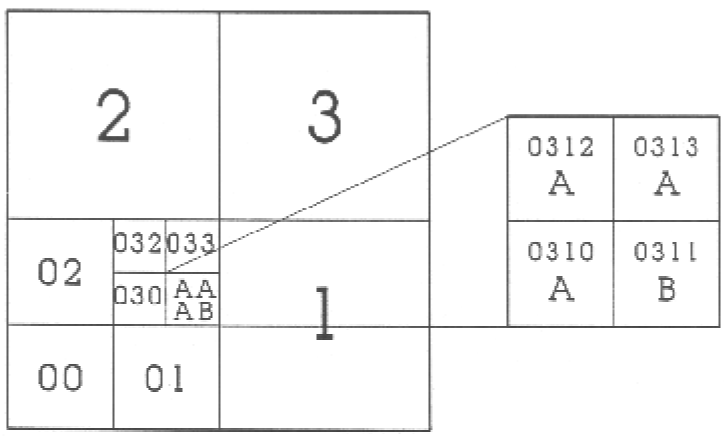
6. Three-dimensional spatial databases are of great interest in geology, geophysics, subsurface hydrology, atmospheric science, oceanography and the mining and oil and gas industries. With reference to one or more of these, discuss the suitability of the octtree as a data model.

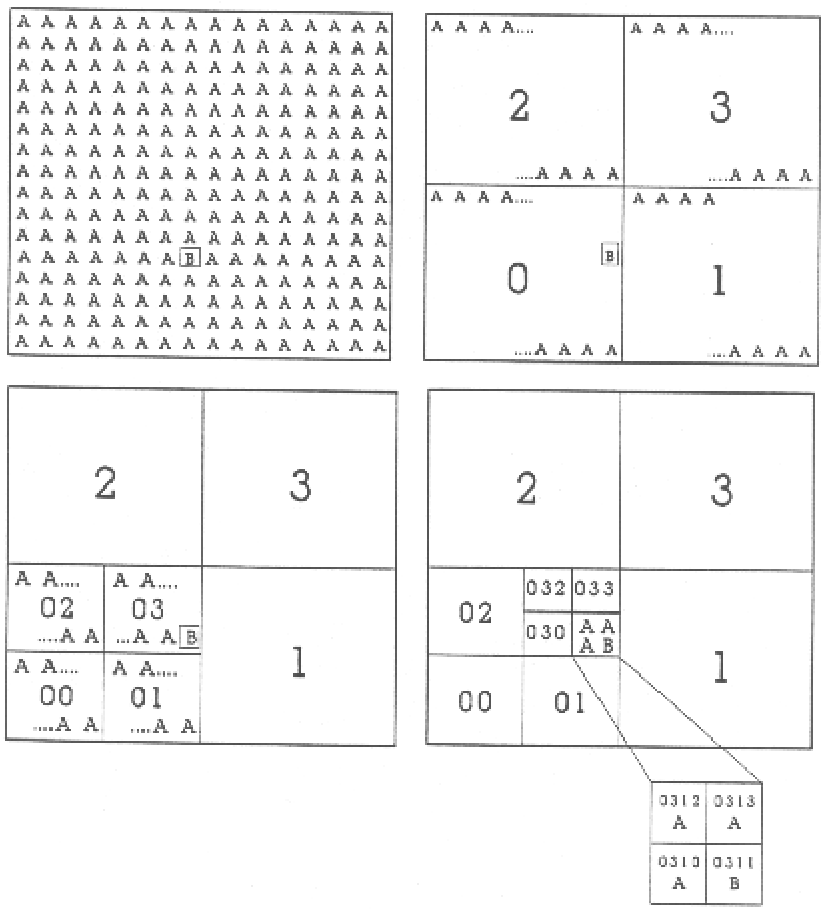
Last Updated: August 30, 1997.

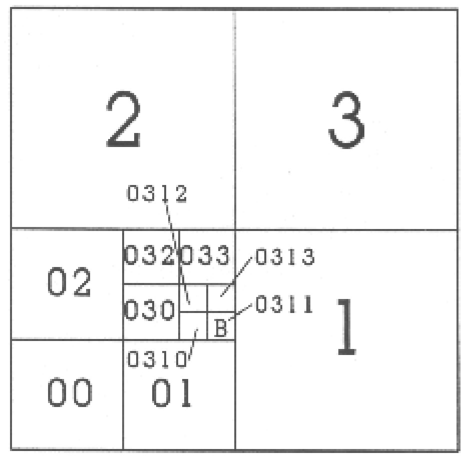
UNIT 36 IMAGES

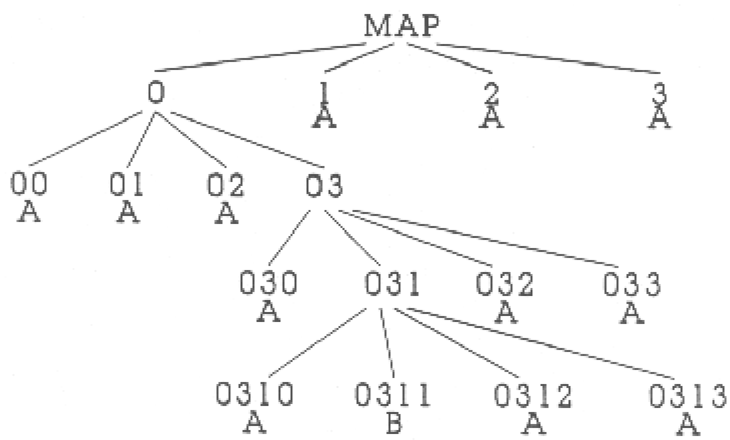


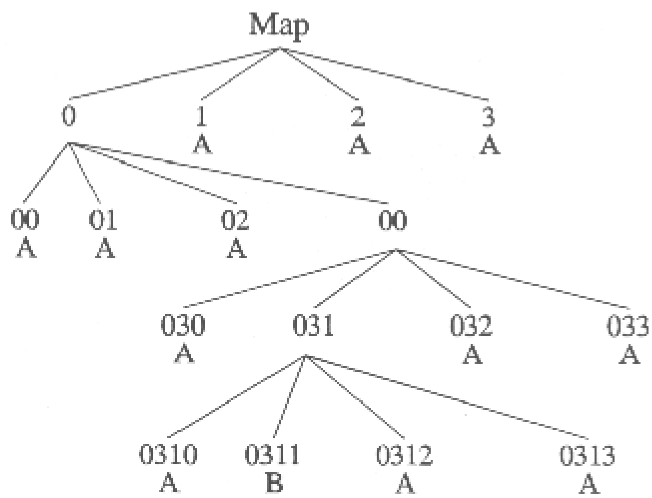

2
3
A A.... A A....
02 03
....A A ...A A B
A A.... A A....
00 01
....A A ....A A
1


2
3
02
032 033
030 AA AB
1
00
01
0312 A
0313 A
0310 A
0311 B


2
3
0
1
02
03
00
01
032
033
030


2
3
0312
032
033
0313
02
030
0311
B
1
0310
00
01


MAP
0
1
A
2
A
3
A
00
A
01
A
02
A
03
030
A
031
032
A
033
A
0310
A
0311
B
0312
A
0313
A



LEVEL

0	pointer			
1	pointer	A	A	A
2	A	A	A	pointer
3	A	pointer	A	A
4	A	B	A	A

Position:	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
Contents:	2	6	A	A	A	A	A	A	10	A	14	A	A	A	B	A	A
(level):	0	1	1	1	1	2	2	2	2	3	3	3	3	4	4	4	4