

LINEAR QUADTREE ENCODING OF RASTER DATA AND ITS SPATIAL ANALYSIS APPROACHES

Sheng Yehua Tang Hong Zhao Xiaohu

Institute of Surveying and Geo-Information Engineering,

China University of Mining & Technology

Xuzhou, Jiangsu Province, P.R. China, 221008

E-mail: shengy2@263.com

Or: shengyh@pub.xz.jsinfo.net

Abstract

This paper aims at two fundamental problems in geographic information system applications that how raster data can be compressed efficiently and how the compressed data is used directly in spatial analysis. Linear quadtree with decimal Morton codes is used to compress raster data into linear quadtree and used in spatial analysis. On the base of summarizing the former researching results, a new linear quadtree encoding method named as Fast Dynamic Linear Quadtree Encoding Method (abbreviated as FDLQE) is put forward, which can construct linear quadtree directly through only once tracing of the whole raster data by using a dynamic stack instead of linear table or one-dimensional array. At the same time, a series of models for direct spatial analysis based on linear quadtree are advanced. These models include geometric features measuring for area region, spatial set operation, and spatial overlay. The testing result on real raster data shows FDLQE is superior to any other linear quadtree encoding approaches in time and memory requirement, and the analysis models are effective.

1. Introduction

Raster data structure expresses spatial objects in the form of grid cells or pixels into a matrix. It is convenient for spatial analysis and it can be easily integrated with DTM and digital remote sensed images in geographical information system (GIS). This structure is one of the effective models to represent spatial objects in GIS. When raster data structure is used, the smaller the size of pixel or grid cell is, the higher resolution of spatial data comes and the more cells with the same attribute value are resulted in, which causes a large amount of redundant information. The memory space required to store spatial data is large. This requirement causes other problems for a GIS. As there are heavy loads for both spatial data processing and communication, much research is concerned with the storage problem. A quadtree is a well-known hierarchical data structure applied to represent geometric data. It is especially useful for raster-based data processing and important for many applications, such as computer vision, computer graphics, image processing, cartography, and so on (Samet, 1990).

The original approach to represent geographical data is the quadtree data structure with pointers (named as Routine Quadtree). The quadtree is represented by a tree whose node is either a leaf or an internal node with four children. The root represents a block and its four sons represent in order of the NW, NE, SW, and SE quadrants. The quadtree is obtained by recursively subdividing the image into quadrants, sub-quadrant, ..., until each block consists entirely of the same value. Each node in the tree needs six fields to maintain the relationship with other nodes. There is one pointer for its father's node and there are four pointers to the nodes of four sons. The latter field holds the data value of this node. A routine quadtree needs much less storage than the original image. But the use of pointers is inevitable, a routine quadtree still consumes a considerable amount of storage space. The memory space can be further saved if a pointer-based quadtree is replaced by a linear coding method (Gargantini, 1982).

A linear quadtree could be regarded as a linear array of certain type of structure elements such that the tree structure is implicitly preserved. The linear structure stores location code and attribute value of any node in a sorted array. Because it needs no pointers to represent the nodes, a linear quadtree is more space saving than a routine quadtree. Linear quadtree could be an effective data structure for raster-based spatial data to reduce the amount of data storage and the requirement of memory and

time during data processing. But how to improve the linear quadtree producing efficiency is still a problem to be studied.

It is a fundamental operation for geographic information system to overlay two maps. The quadtree data structure has been used as the underlying representation for maps in several experimental GIS successfully. Once raster data is encoded, it is difficult to be processed directly like a matrix when carrying out spatial analysis. The usual method is decoding the compressed data to restore its original matrix, then carrying out further analysis. This process is also time and memory consumptive.

How raster data can be compressed efficiently and how the compressed data is used directly in spatial analysis are two hot problems to be solved in GIS studies. In this paper, we propose a new algorithm named as Fast Dynamic Linear Quadtree Encoding (FDLQE) which can produce a linear quadtree through scanning the image only once. A series of models for spatial analysis based on linear quadtree encoded raster data are also advanced.

2. Review of the Linear Quadtree

The fundamental principle of quadtree encoding is that recursive decomposition is applied to partition an image or raster-based data layer with the size of $2^n \times 2^n$ ($n > 1$, and n is the resolution factor of the image) into four sub-images with the same size. Every part is denoted as a node. If all pixels have the same attribute value in one sub-image, this sub-image need not partition any more, and it is denoted as a leaf. The sub-image is not further partitioned into four sub-images until all pixels in any sub-image are of the same value. Eventually, the raster data is organized as a inverted tree (shown as Fig.1). This quadtree constructing process is names as top-to-down partition method. Because this process must repeatedly check whether all the pixels in any sub-image are of the same value and decide whether the sub-image needs be portioned ulteriorly, this process is time consumptive when the original raster data is with large size. Further more, to represent this quadtree, at least 5 additional pointers that have no matter with the location of the nodes or leaves are used, which take up nearly 90% of the memory space used by a quadtree (Stewart, 1986).

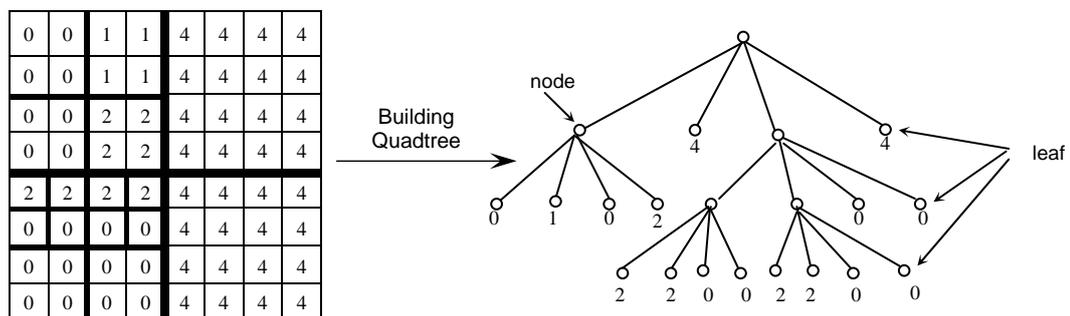


Fig.1 Partition Process and Organizing Structure of Quadtree

Gargantini (1982) advanced an efficient way to represent quadtrees, named as linear quadtree. The concept of linear quadtree is proposed to eliminate pointers such that the memory space occupied by pointers is saved. It can be regarded as an encoding scheme with the following characteristics. (1) Only leaves are stored. (2) The path from leaf to node is reserved in its code implicitly. (3) Codes are arranged in a strictly increasing sequence. Morton codes are used to represent the location of the node or leaf. Principally, a unique Morton number is assigned for each sub-image with uniform value. Four quadrants along a path from a parent node are visited and encoded as a weighted code 0, 1, 2 or 3, corresponding to the directions of northwest, northeast, southwest and southeast. The number of digits in each node or leaf indicates how many times the subdividing is done. Thus, the weight 4^{n-i} , $1 \leq i \leq n$, each digit identifies a situation in which the node or leaf belongs to the i th decomposition. Traditionally, down-to-top merging algorithm is used to construct linear quadtree. The raster data is scanned with breadth-first traversing, and the Morton code of each pixel is calculated. Once the raster data is scanned and quaternary codes are assigned to all pixels, those quaternary codes are sorted, and both the codes of a node and its corresponding attribute value are stored in an array or list. The quadtree is therefore linear because codes for each pixel are recorded in a linear array. After all pixels are assigned quaternary codes, a condensation procedure is applied to merge neighboring pixels whose attribute value are the same and Morton codes have the same representation except for the last digit. For each successful merge, four quaternary codes are eliminated from the

list and a new quaternary code is appended with a special mark to replace them. A final quadtree is derived until the condensation procedure is recursively applied to merge all neighboring regional pixels. All operations on the quadtree are then manipulated on a list or table rather than in a sequence following the order of pointers. The 8×8 raster data shown in Fig.1 can be finally represented as (000,0), (010,1), (020,0), (030,2), (100,4), (200,2), (201,2), (202,0), (203,0), (210,2), (211,2), (212,0) (213,0), (220,0), (230,0), (300,4).

Although each digit of quaternary code can express the spatial relationship and the location of the quadrant, quaternary code does not accord with thinking of mankind and it is not supported by most computer systems. For the purpose of convenience, decimal Morton code (abbreviated as MD) based on natural number is used to express the location of the leafs in quadtree. Decimal Morton codes is used to assign consecutive numbers, from 0 to $2^n \times 2^n$, starting from the top left pixel to the bottom right pixel in a two-dimensional raster data as shown as Fig.2. The raster data shown in Fig.1 can be organized in a list or a table likes: (0,0), (4,1), (8,0), (12,2), (16,4), (32,2), (33,2), (34,0), (35,0) (36,2), (37,2), (38,0), (39,0), (40,0), (44,0), (48,4).

	X								
	0	1	2	3	4	5	6	7	8
0	0	1	4	5	16	17	20	21	64
1	2	3	6	7	18	19	22	23	66
2	8	9	12	13	24	25	28	29	72
3	10	11	14	15	26	27	30	31	74
4	32	33	36	37	48	49	52	53	96
5	34	35	38	39	50	51	54	55	98
6	40	41	44	45	56	57	60	61	104
7	42	43	46	47	58	59	62	63	106
8	128	129	132	133	144	145	148	149	192

Fig.2 Decimal Morton Code Assignment

The constructing of linear quadtree based on MD code can also use the bottom-to-top merging procedure. Before creating linear quadtree, a one-dimensional array should be defined at the same size of the pixel numbers in the original raster data (to be exact, $2^n \times 2^n$). Breadth-first scanning is carried to calculate the MD code MD_{ij} of every grid cell, and the attribute value of the corresponding cell is stored directly in the array according to its MD code. Because the location of any cell in the array is nicely the decimal Morton code in its increasingly sorted location, the sorting procedure can be bypassed. Besides, the subscript itself of the array is the MD code of the cell, any other memory space is not needed to store the MD code. After the array is filled up, a condensation process is applied to merge the adjacent cells with the same attribute value, just like that of linear quadtree with quaternary codes. There are at least three methods to calculate the MD code of any grid cell (Gong, 1993). (1) Quaternary code is produced first for every cell, and then the quaternary code is changed into MD code. (2) The column and row numbers of a cell are first transformed into a special pseudo codes, the MD code is calculated from these pseudo codes. (3) Bit operation is used to create MD code directly.

The major problems are that the requirement of memory size is proportional to the resolution of a given raster data, the MD code calculating of each grid cell is time consumptive, and the merging procedure is difficult. The proposed Fast Dynamic Linear Quadtree Encoding (FDLQE) approach can remedy these scarcities.

3. Fast Dynamic Linear Quadtree Encoding of Raster Data Based on Decimal Morton Codes

The FDLQE method can construct linear quadtree through only once scanning the whole raster data. It makes use of a dynamic stack instead of a linear table or an array, and the characteristic of stack that who entered first emerges finally. While the attribute value of a cell is extracted from the original raster data in the order of MD code, it is pushed into the stack and

recorded as a dynamic record. At the same time, some rules are used to condensing the stack. After all the cells in the raster data are traversed over and the records in the stack can not be condensed, the records remained in the stack are just the linear quadtree encoding result. The FDLQE approach includes 3 steps. (1) Extracting grid cells from the original raster data in the order of MD code. (2) Comparing the attribute values of neighboring cells and pushing into stack. (3) Condensing the stack. These steps are illustrated in Fig.3.

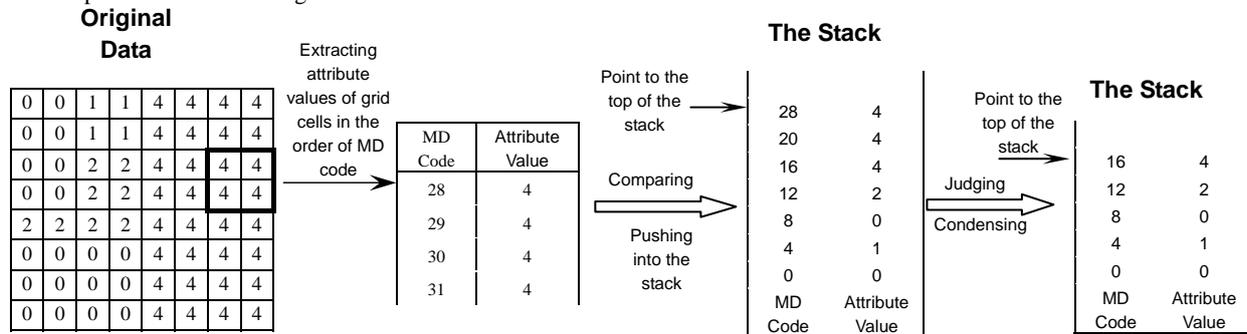


Fig.3 FDLQE Processing Steps for an 8x8 Raster Data

Extracting Grid Cells

Because the MD code of a raster data is just the decimal normal number in its increasing order (i.e. from 0 to $2^n \times 2^n - 1$), the normal number can be regarded as MD code of any grid cell and used as the scanning base. Since the MD code is the result from interleaving combination of row and column number of a grid cell in their binary format (Gong, 1993), the corresponding binary row and binary column number can be decomposed by using bit operation from the MD code or normal number which is changed into binary format. This method is shown as Fig.4. The decimal formats of the decomposed binary row number and binary column number are what are needed to read the attribute value of a cell. With this method, the attribute values of 4 consecutive grid cells are read from the original raster data in the order of MD code or normal number.

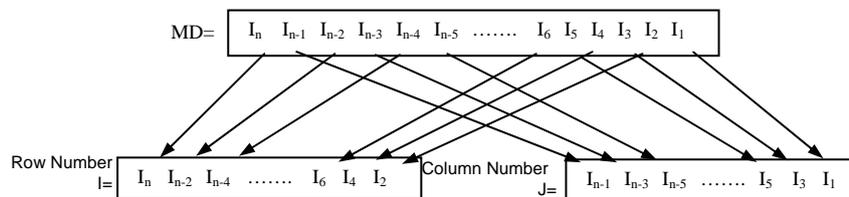


Fig. 4 Row and Column Numbers Decomposing from MD Code

Comparing the attribute values of neighboring cells and pushing into stack

The read 4 sequential grid cells are compared whether their attribute value are equal. If they are the same, the MD code and its corresponding attribute value of which cell gets the lowest MD code in the 4 sequential grid cells are pushed into the stack. Otherwise, all the MD codes and their corresponding attribute values of the 4 cells are pushed into the stack in the increasing order of the MD codes. And the pointer to the top of the stack is recorded.

Condensing of the Stack

The fundamental consideration of FDLQE is that the stack is used to dynamically merge the records to upper nodes to construct linear quadtree. When the number of records in the stack is more than or equal to 4, it must judge whether these records can be merged to upper node. The MD code of the top record in the stack is used to decide whether the stack must be condensed and how many times the record can be merged. The times T that the record should be merged is calculated with equation 2, if the MD code of the top record satisfies equation 1.

$$\text{mod}\left[\frac{(MD+1)}{4^{K+1}}\right] = 0 \quad (1)$$

$$T = \max(K) \quad (2)$$

Where *mod* is seeking remaining function. The possible integral solutions of *K* are calculated first, and then *T* is the maximum number of *K*s. In the fact, *T* needs not be calculated with these equations. After the decimal number of MD+1 is changed into its corresponding quaternary value, the count of zeros at the end of this quaternary value is just *T*-1.

A variable *L* is taken as a recursive variable to control the condensing procedure with the value from 1 to *T*. In every recursive, the MD codes of 4 successive records on top of the stack are compared whether their sequential difference are equal to 4^L . If these relationships do not exist, the records in the stack are leafs and the stack cannot be condensed, the merging circulation should be exited. Otherwise, the attribute values of these records are further compared whether they are equal. If they are not equal, the records are leafs too, quit from the recursive. If they are the same, which means that 4 nodes in this hierarchical level should be merged, and the point to the top record in the stack is moved to the record whose MD code is the minimum, the other 3 records on top of this record are kicked out from the stack. The recursive condensing procedure goes on until *L* is increased to *T*.

Procedure of FDLQE algorithm

The entire algorithm of FDLQE method can be expressed in a flowing chart shown in Fig.5.

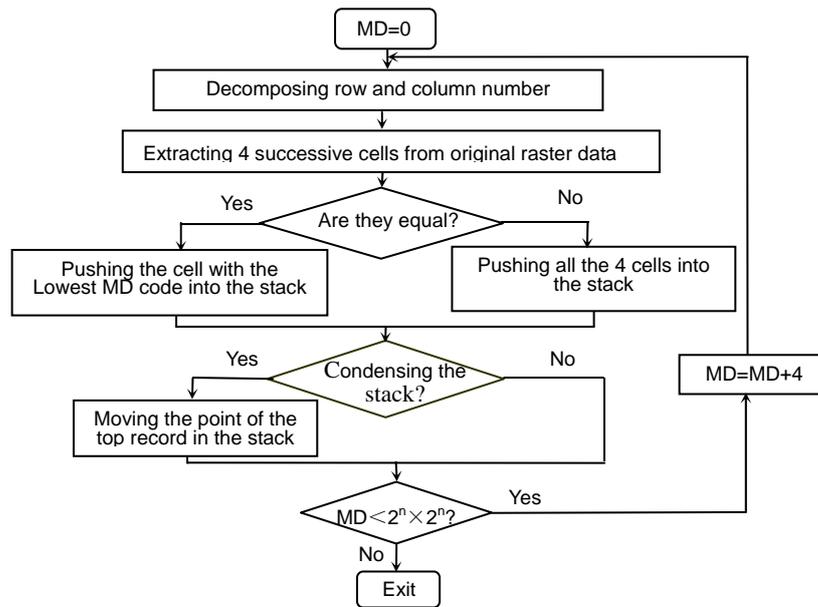


Fig.5 Flowing Chart for FDLQE Algorithm

4. Spatial Analysis of Raster Data Based on Linear Quadtree

Spatial analysis is one of the basic functions of geographic information system. To be an important data model, raster data is convenient for spatial analysis such as overlay. Once raster data is encoded in its linear quadtree, it is difficult to be processed directly like a matrix when carrying out overlay. The normal method is decoding the encoded data to restore its original array, then carrying out further analysis. This process is also time and memory consumptive. Therefore, Some spatial analysis algorithms based on linear quadtree encoded raster data should be put forward, such as overlay, set operation, geometric feature measuring and so on.

Geometric Features Measuring for Area Region

In a linear quadtree encoded raster data, every leaf node is a homogenous square block corresponding to an area region. The MD code of this block corresponds to the location of the top left grid cell, and the difference (denoted as *V*) between this MD

code and the immediately adjacent MD code denotes the size of this block. The row number I and column number J of the top left cell can be decomposed from the MD code with the method mentioned above. The width of this block is $L_x=V^{1/2}dx$, and the height of this block is $L_y=V^{1/2}dy$. They area of this region is $S=V \cdot dx \cdot dy$, and the perimeter of this block is $P=2\sqrt{V} \cdot dx+2\sqrt{V} \cdot dy$, where dx and dy represent the traverse width and longitudinal width of the grid cell respectively.

Set Operation of Linear Quadtree

Set operation is the foundation of spatial query (Huang, 1995). Set operation of linear quadtree is based on Boole's Logic algorithms. The basic set operations for raster data include set union, intersection and difference. At first, the set operations of two basic blocks should be constructed. Supposing two basic blocks with their MD codes MD_A and MD_B , then their union intersection and difference can be carried out by:

$$MD_A \cap MD_B = \begin{cases} MD_C = 0, MD'_C = 0, & \text{when } MD'_A \leq MD_B \text{ or } MD'_B \leq MD_A \\ MD_C = \max(MD_A, MD_B), MD'_C = \min(MD'_A, MD'_B), & \text{otherwise} \end{cases} \quad (3)$$

$$MD_A \cup MD_B = \begin{cases} MD_C = \{MD_A, MD_B\}, MD'_C = \{MD'_A, MD'_B\}, & \text{when } MD'_A \leq MD_B \text{ or } MD'_B \leq MD_A \\ MD_C = \min(MD_A, MD_B), MD'_C = \max(MD'_A, MD'_B), & \text{otherwise} \end{cases} \quad (4)$$

$$MD_A - MD_B = \begin{cases} MD_C = MD_A, MD'_C = MD'_A, & \text{when } MD'_A \leq MD_B \text{ or } MD'_B \leq MD_A \\ MD_C = 0, MD'_C = 0, & \text{when } MD_A \leq MD_B \text{ and } MD'_A \leq MD'_B \\ MD_{C1} = MD_A, MD'_{C1} = MD'_B - 1, MD_{C2} = MD'_B + 1, MD'_{C2} = MD'_A, & \text{otherwise} \end{cases} \quad (5)$$

Where MD' means the MD code minus 1 of the immediately adjacent leaf node or the bottom right cell of this block. The set operation of two linear quadtree encoded raster data is shown as Fig.6.

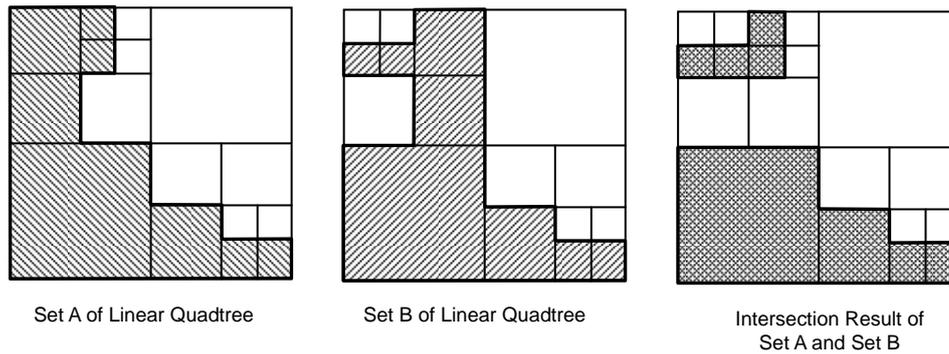


Fig.6 Example of Intersection of Two Linear Quadtree

Spatial Overlay of Linear Quadtree

Spatial Overlay uses at least two layers of geographical element to create a new theme. The result of overlay is a new data file convenient to geographic analysis. For example, two overlaid layers of county boundary and land use can be used to calculate the area of one county. When raster data format is used in overlaying, the two layers should be at the same size. Supposing two layers of raster data to be overlaid have been encoded into linear quadtree, and the data for these layers are:

$$\text{Layer1} = (MD_i, Attr_i), \quad i=1,m \quad (6)$$

$$\text{Layer2} = (MD_j, Attr_j), \quad j=1,n \quad (7)$$

Then the procedure of spatial overlay with these two layers can be express as the following steps:

- (1) Open the two linear tables of these two linear quadtree encoded layers of raster data, and create a new linear table to record the MD code and attributes of these two layers.
- (2) Detect the minimum MD code from the remaining records in both these two tables.
- (3) Decide the corresponding attribute values of this minimum MD code on each layer, record the MD code and the attributes of these two layers in the newly created table to form a new record, and expurgate detected record from the original two linear tables.
- (4) Repeat step 2 and step 3 until all the records in both linear tables are recoded in the newly created table.

This overlay approach can overlay more than two layers of raster data at the same time. After these layers are overlaid, the attributes of each record can be used to calculate new attribute value with appropriate mathematical models. Thus a new geographic theme results in.

5. Discussions and Future Research

In order to check the validity of the proposed algorithms and models, all of them are realized on a PC platform with VC++ programs. Actual raster data layers with the size from 256×256 to 2048×2048 are used in the test.

It is proved that FDLQE algorithm is more time and memory saving than any other linear quadtree encoding methods, and larger the size of raster data is the more superiority of FDLQE is. The high performance of FDLQE is because that: (1) it needs no any sorting process to the records in the stack, (2) it traverses the whole raster data in the order of MD code only once, (3) it extracts 4 successive cells in every circulation, and the cell with the lowest MD code requires changing MD code into row and column number, (4) it uses a dynamic stack instead of a fixed table or array, records are recursively merged to upper nodes during the traversing, it needs not to store MD codes and attribute value of every cells, and it needs no special memory block to store encoded linear quadtree.

It is convenient to directly carry out spatial analysis with linear quadtree for raster data such as set operation, spatial overlay, spatial clustering and so on. However, some other special spatial analysis directly with linear quadtree encoded raster data like spatial query, counter tracing, buffering, classifying should be further researched in the future.

Acknowledgement

This research was supported by National Natural Science Foundation of China, under item serial number of 49871069. This support is heartily acknowledged.

References

1. Samet, H., 1990, Applications of Spatial Data Structures, Addison-Wesley, New York
2. Gargantini, L., 1982, An Efficient Way to Represent Duadtrees, *Communications of ACM*, 25: 905-910
3. Gong, J., 1993, Data Organization and Processing Approaches for Unitary Spatial Information System (In Chinese), Press of Wuhan Technical University of Surveying and Mapping
4. Chang, H., Liu, S. Tso, C., 1997, Two-Dimensional Template-Based Encoding for Linear Quadtree Representation, *Photogrammetric Engineering & Remote Sensing*, 63(11): 1275-1282
5. Lin, T., 1997, Set Operations On Constant Bit-Length Linear Quadtrees, *Pattern Recognition*, 30(7): 1239 -1249
6. Sheng, Y., Tang H., Du, P., 2000, Fast Dynamic Encoding of Linear Quadtree and Its Realization, *Journal of Wuhan Technical University of Surveying and Mapping* (In Chinese), 25(4): 324-328
7. Tan, G., Lin, Z., 1995, An Optimal Natural Digits Based Linear Quadtree Construction Algorithm, *Acta Geomatica et Cartographica Sinica* (In Chinese), 24(3): 204-210
8. Ji, Y., Ding, W., Chen, Guo-liang; Gui, X., et al, 1999, Parallel Algorithms of Linear Quadtree on MPP, *Chinese Journal of Advanced Software Research*, 6(1): 95-100
9. Yannis, M., Enrico, N., Apostolos, P., et al, 1997, MOF-tree: A spatial access method to manipulate multiple overlapping features, *Information Systems*, 22(8): 465-481
10. Chang, H., Cheng, L., Liao, S., et al, 1996, Breadth First Linear Quadtrees for Water Resource Management in Geographical Information Systems, Application of Geographic Information Systems in Hydrology and Water Resources Management Proceedings of the HydroGIS'96 Conference, Apr 16-19, 1996, Vienna, Austria, IAHS Press Wallingford, 55-62
11. Steward, I., 1986, Quadtrees: Storage and Scan Conversion, *Computer Journal*, 29: 60-75
12. Huang, C., Chung, K., 1995, Fast Operations on Binary Images Using Interpolation-Based Bintrees, *Pattern Recognition*, 28(3): 409-420