A UNIFIED DATA STRUCTURE TO REPRESENT VECTOR AND RASTER DATA IN GIS

Li Deren Gong Jianya
Department of Photogrammetry and Remote Sensing
Wuhan Technical University of Surveying and Mapping
Wuhan, China

ABSTRACT

In the literature a number of proposals have been made for an unified data structure to represent vector and raster data. The unified data structure is based on linear quadtree encoding and multi-grids technique, and it can combine the advantages of vector and raster structures and support an integrated system to contain GIS, DTM and RS.

In this data structure, all geometric data are expressed by the address keys of linear quadtrees instead of X and Y coordinates. In order to improve the representation precision of raster, an approach for fine-dividing grids is proposed, on the other hand, in order to index database, an indexing file based on linear quadtree is used. The location of a point is represented by two Morton keys, one of them indicates its location in the basic grid, and the other one corresponds to the fine-divided grid. A line feature is described with a set Morton keys, which contains not only the end points and sampling points but also consists of the whole route that passes the basic grids. Similarly, a surface feature consists of the borders similar to the line representation and the whole terrain area encircled by its borders. The surface features as coverage data can be stored as a two-dimensional run-encoding file. These geometric data can be considered as vector form or raster form or both.

Key Words: Linear quadtree, Raster, Vector, Unified data structure, Two-dimensional run-encoding.

1. INTRODUCTION

There are two kinds of data structures: vector-based and raster-based, in the conventional geographical information system. The storage of data in a vector-based form is superior to the storage of that in raster form for several reasons. In the vector-based data structure the information stored follows the original data and the degree of approximation used in storing those data is controllable; generally, one piece of vector data may be used in representation of more than one features (for instance, a wall may be part of house and also the edge of a road), this contrasts with raster data, where each pixel may be shared by several features, whereas in a raster data overlay, each pixel can represent only one kind of feature; there are also problems of resolution with raster data, a line of length less than the pixel size being indistinguishable from a point in the raster image; the vector data are more compact than raster data, as the white space of the map need not be represented; the vector structure has the better topological representation than the raster one.

A major disadvantage of vector data with respect to raster data is that they do not contain explicit two dimension relationships. To do an operation such as ' Finding all features near this roadway in 5 kilometers' can require a great deal of work. To do the set operations (intersection, union, subset) will be very difficult in the vector data structure. The raster data structure can just overcome the disadvantages of the vector data structure. On the other hand, the combination between remote sensing and GIS is on the increase. In order to do the geographical analysis by combining the remote sensing data in raster-based form with conventional GIS in the vector-based form, it is necessary to convert the vector data in GIS into raster data.

A multiple functions GIS is often required to do convertion between vector data and raster data. Thus, to find a data structure containing the vector and raster data becomes increasingly interesting. It can combine the advantages of the vector and the raster data structures. Some schemes for the integration of the two kinds of data has been investigated (B. Sonne & B. Zillien 1990, M. Molenaar & D. Fritsch 1990).

In the geographic information system, a necessary evolution is the integration of it with remote sensing (Ehlers et al. 1989). There are three kinds of integration strategies, which are defined as "separate but equal", "seamless integration", and "total integration". The total integration is the highest level one and the long-term goal. In this kind of system, a very important thing is to design an unified data structure to represent vector and raster data and an integrated database to combine attribute, geometry, DTM and RS.

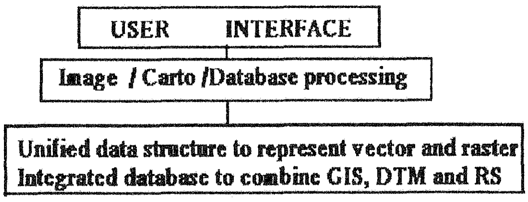

Figure 1 Total integration of GIS, DTM and RS

Currently, the quadtree has received considerable attention as a data structure for GIS application (Mark and Lauzon 1989, Yong Hongguang 1990, Molenaar and Fritsch 1990). Quadtrees appear to have many advantages for handling coherent spatial data, and especially good for geographical overlays.

This paper first introduces linear quadtree and two-dimensional run-encoding. A strategy to improve the representation precision of raster and an indexing method based on linear quadtree are presented in section 3. In the section 4, the data structures of three typical features are discussed. Finally section 5 draws some conclusions.

2. LINEAR QUADTREE AND TWO-DIMENSIONAL RUN-ENCODING

There are two methods for compacting raster data: quadtree and run-length encoding. But here the two-dimensional run-encoding is different from the conventional run-length encoding. It is a space-efficient form of linear quadtree, and uses the same address keys as linear quadtree.

2.1 Linear Quadtree

The linear quadtree is a data structure which is based on the regular decomposition of a square into quadrants and sub-quadrants (see Mark 1989). Such structure represents only leaf nodes, with nodes identified by numeric keys. The form of these keys permits topological and spatial relations to be determined from the key values. The data structure is conceptually then a list of the leaf nodes in sequence by key. In general, Morton keys are used as the numeric keys (Figure 2).

7	42	43	46	47	58	59	62	63
6	40	41	44	45	56	57	60	61
5	34	35	38	39	50	51	54	55
4	32	33	36	37	48	49	52	53
3	10	11	14	15	26	27	30	31
2	8	9	12	13	24	25	28	29
1	2	3	6	7	18	19	22	23
0	0	1	4	5	16	17	20	21
	0	1	2	3	4	5	6	7

Figure 2. Morton keys

There are several ways to obtain Morton keys, while bit interleaving is the fast one (Gong 1991). Bit interleaving consists of taking the bit representations of the row and the column number, and forming a key consisting of alternating bits from each number (Figure 3).

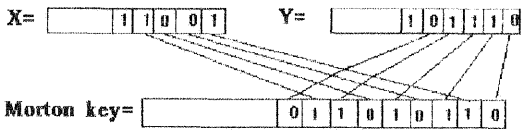

Figure 3. Generation of Morton key

A "bottom-up" quadtree construction may be the easier and faster than the "top-down" construction (Gong 1991). Appropriate sets of four mutually-adjacent cells can be examined and combined into a node of level 1 if they all have the same value. Then, if four such level-1 nodes have the same value, they are combined into a level-2 node and so on (Figure 4b). This method will lead to the same set of quadtree nodes as the top-down approach.

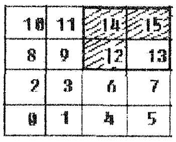

Fig. 4a

Linear quadtree

M1	Value
0	0
4	0
8	0
12	1
13	0
14	1
15	1

Fig. 4b

2DRE

M1	Value
0	0
12	1
13	0
14	1

Fig. 4c

Figure 4. Linear quadtree and 2DRE

2.2 Two-dimensional Run-encoding

Two-dimensional run-encoding (2DRE) is an approach for the compact representation and manipulation of linear quadtree (Lauzon 1983, Mark and Lauzon 1989). A 2DRE file can be constructed by run-encoding a linear quadtree, that is, by combining runs of consecutive leaves of the same value into a single data element. In figure 4b, the three values of the former leaf nodes are the same, and they are combined to form a record (Figure 4c).

Two-dimensional run-encoding exhibits several desirable features for a data structure for use in geographical data processing; for example, multiple attribute encoding, random access and error recovery. This enhances the performance of GIS procedures such as search and overlay, and result in greater storage efficiency and flexibility for processing of geographical information than do other quadtree structures.

3. MULTI-GRIDS TECHNIQUE

The original cells of raster serve basic grids. To improve the representation precision of raster, some particular basic grids containing points and lines are divided into many fine grids, and to index the 2DRE file, basic grids are grouped into rough grids as indexing blocks.

3.1 Fine-divided Grids for Improving Representation Precision of Raster

A main disadvantage of raster structure is too low accuracy. In this section, an approach by using fine-dividing grids is presented. In figure 5, each basic grid is encoded in Morton key(M1). In order to improve the geometric resolution of a raster, those basic grids that contain points and lines are redivided into 256*256 fine-grids, which are also encoded in Morton key(M2). Thus, a pair of coordinates x, y can be converted into two Morton keys M1 and M2, and replaced by them. In this way the represented precision of raster data for points and lines will be increased by about 256 times, however, the storage is increased only by 2 bytes for each sampling point and intersection point between a line and a basic grid side.

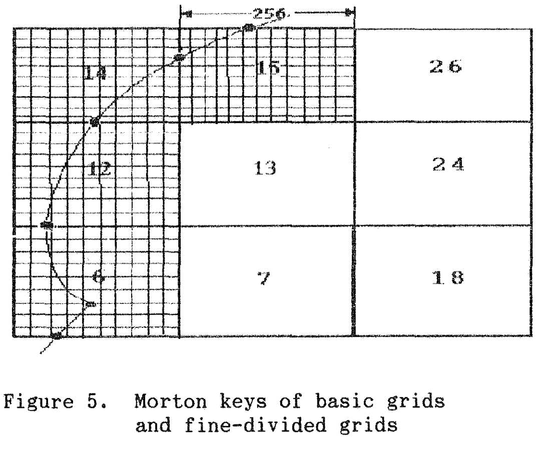

Figure 5. Morton keys of basic grids and fine-divided grids

3.2 Rough Grids for Indexing 2DRE File

It is often necessary to access only a part of the spatial database, that is, to index into the database and retrieve only data pertaining to a particular area. B^{++} tree is a popular approach in a general database. However, B^{++} trees do not map directly into the quadtree structure, that is, there is no logical correspondence between the two structures. Accordingly, a block of the B-tree covers an irregular varying unit of space within the quadtree.

To overcome this problem, a quadtree indexing method is proposed, where the index corresponds to some higher (non-leaf node) level of the quadtree and so gives a regular division of space, which is termed rough grid and is also encoded in Morton key(M0). A limitation when generating 2DRE is that a "leaf node" can not exceed a rough grid. The index is also stored as a linear list in Morton key. The sequences of the records in the index file implicitly represent Morton keys(M0) so M0 can be not written in the index file. Each record of the index contains only the number of bytes from the start of the file (Figure 6).

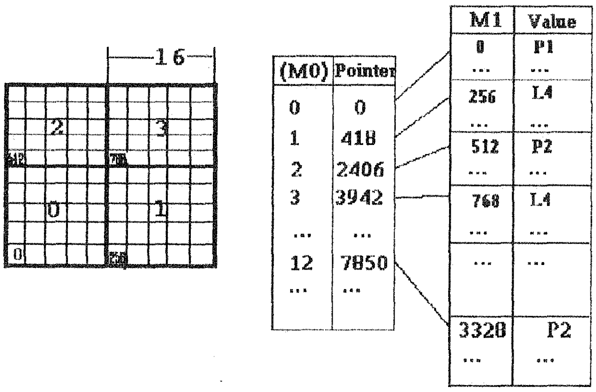

Figure 6. Rough grid index for 2DRE

The index readily provides random access into a "leaf node" of 2DRE. to retrieve data at particular point, the Morton key(M1) of the point in the basic grid is divided by the rough grid size, and a Morton key(M0) in rough grid is obtained. This gives the number of the relevant index record, from which the number of bytes can be used to retrieve the relevant point portion of the 2DRE. This portion is then searched linear until the required point is found.

4. UNIFIED DATA STRUCTURES OF FEATURES

There are three types of features including points, lines, and surfaces. Several files may exist, each of them contains information on a primitive element or a kind of feature.

4.1 Data Structure of the Point Features

A point feature may simultaneously be an intersection of several lines. Therefore we treat the point features and the intersections of lines as the same objects, which are called nodes and stored in the same data file.

A node can be geometrically described only by its location. Node data would be held as files in which each node was represented by two Morton keys M1 and M2 of the node, M1 and M2 represent respectively the Morton key of the basic grid and that of the fine-grid in which the node is located. Each record in the node file would contain a node identifier NODE_ID, two Morton keys M1 and M2 (see figure 7).

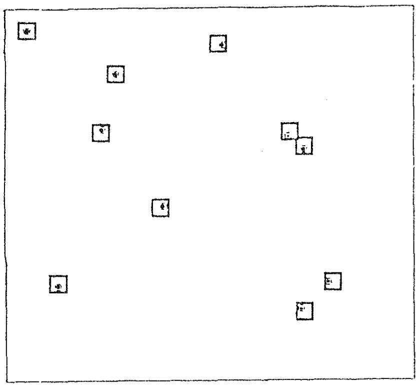

Figure 7. Structure of points

4.2 Data Structure of the Line Features

A line feature has a shape and should be described by the whole route of the line. It is suitable that line features are treated as vector structures. But in order to interface raster data, the vector structures are also represented by linear quadtree address keys. The coordinates x, y are converted into Morton keys M1 and M2. It is noted that the converted coordinates not only are those of the original sampling points but also include those of the intersection points between the line and each basic grid side(Figure 8).

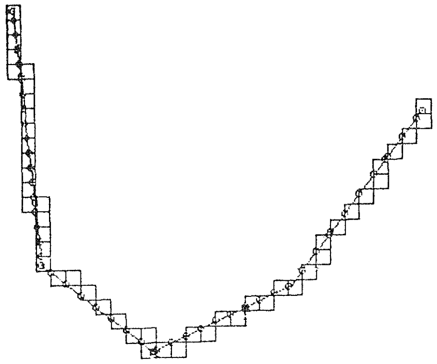

Figure 8. Structure of ARC

To consider a line feature may be divided into several arcs, an arc file should first be first created. Each arc uses a record in the arc file, which contains an arc identifier ARC_ID, two NODE_IDs of the start-node and the end-node of the arc, a string of Morton keys M1 and M2 of medium points.

A line feature consists of arcs. Its data structure is linked with the file of the arcs. The structure contains a line identifier LINE_ID and some ARC_IDs of arcs composing the line feature.

4.3 Data Structure of the Surface Features

A surface feature involves its borders and all the cells bound by its borders. The data structure of the borders is similar to that of lines, while the cells are constructed by two-dimensional run-encoding(Figure 9). An overlay is represented by a 2DRE file. Because the 2DRE of an overlay can occupy a large file, an index file is necessary.

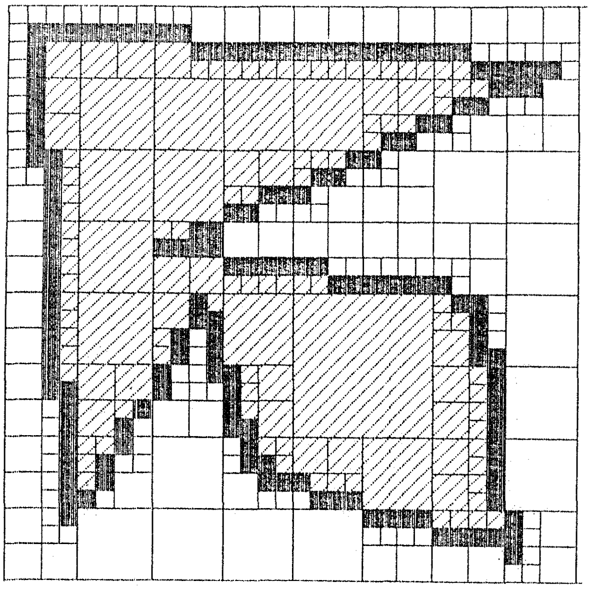

Figure 9. Structure of surface

In the 2DRE file, each "leaf node" uses a record, which contains only one Morton key M1, and a cycle pointer instead of the attribute/value of the "leaf node" in figure 4c. A purpose to set up pointers is to build the object-oriented data models. All leaves that belong to a surface feature are linked by the chain pointers, and connected with the data file of the surface features.

The data structure of surface features should contain both the border information and the cell information. A surface feature uses a record which contains a surface identifier SURFACE_ID, some ARC_IDs of the boundary arcs and a pointer to point at the first "leaf node" record of the feature in the 2DRE file.

4.4 Data Structure of the Complex Features

A feature consisting of several features is called a complex feature. The data structure of complex feature contains a complex feature identifier COMPLEX_ID and some identifiers of its component features, which may be different types of features.

4.5 Data Model Based on the Unified Data Structure

The concepts of object-orientation can be employed to build the geometric data models. In geometry, there are only three types of basic features and several primitive elements such as node, arc and "leaf node" in 2DRE. A feature consists of one or more primitive elements, more than one features may compose a more complex objects. Some underlying data are propagated from the components to the composite object in terms of the identifiers or the pointers. On the other hand, aggregation would be helpful to share the common objects, which could establish the topological relationships among objects. The data models including the primitive elements, the simple and the complex features are shown in the following diagram.

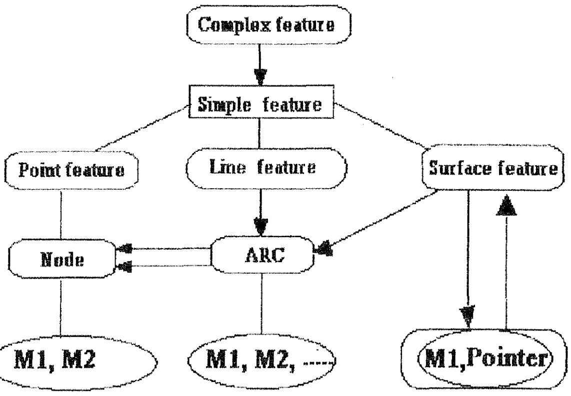

Fig. 10. Data model based on unified data structure

The arrows in the diagram express the identifier linking or the pointers in the data files, the ellipses contain the most primitive data in GIS. The diagram shows that the vector structures and the raster structures do not have any obvious distinctions because they use the same primitive data format--Morton key, which can guarantee the unified data structure serve the two kinds of data.

5. CONCLUSION

This paper has discussed the linear quadtree and two-dimensional run-encoding, fine-divided grids for improving the representation precision of raster and rough grids for indexing 2DRE file. A modeling of an unified data structure and a data model based on the structure has been proposed for future GIS. The data structures based on linear quadtree has been formed by using the object-oriented programming language C^{++}. The data models according to Fig. 10 have been built by using the same language. It has illustrated that the structures are powerful for analysing and processing the geoinformation. Some algorithms based on the data structure, such as spatial querying, automatically establishing topological relations among features, directly converting vector data into linear quadtree and the linear quadtree into vector data, are developed. Here are some examples for statial analysing and data processing: figure 11 is an example for querying all susfaces neighbouring the the Surface A; figure 12 is an example for querying all surfaces relating to the Line A; figure 13 is a result of an intersection set operation; figure 14 is an example converting vector into linear quadtree; and figure 15 is a vector map from linear quadtree. The database with the unified data structure may need more storage space than the single vector structure, but with the development of the hardware it is not a crucial issue.

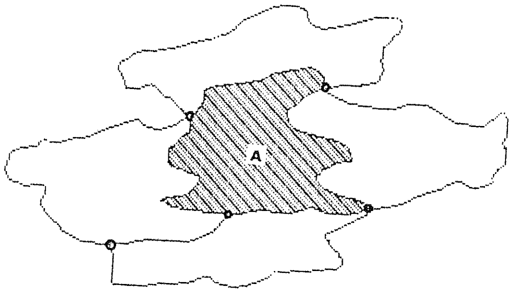

Fig. 11 Querying all susfaces neighbouring the the Surface A

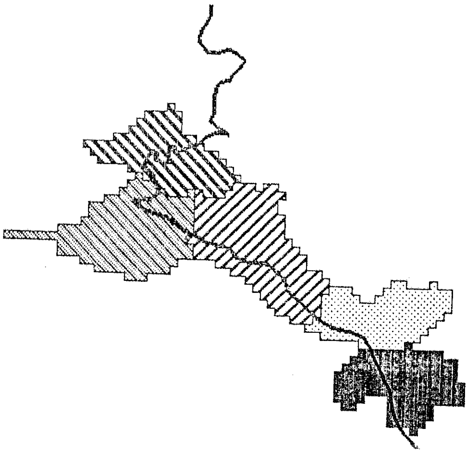

Fig. 12 Querying all surfaces relating to the Line A

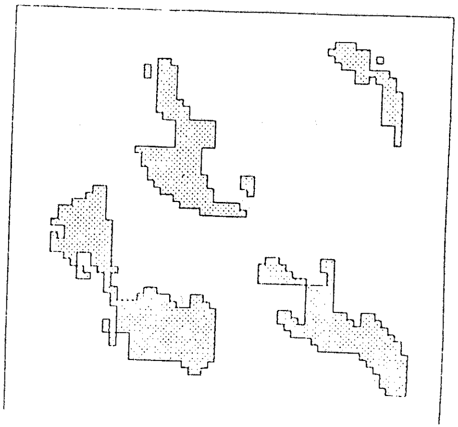

Fig. 13 The result of an intersection set operation

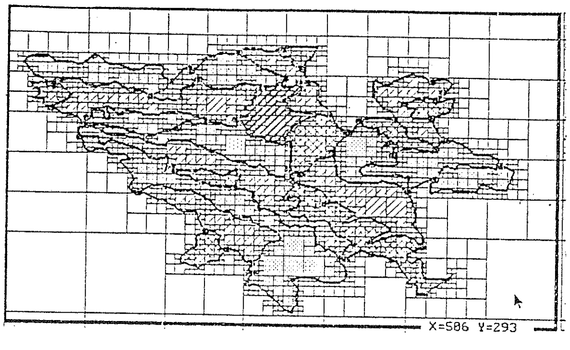

Fig. 14 Converting vector into linear quadtree

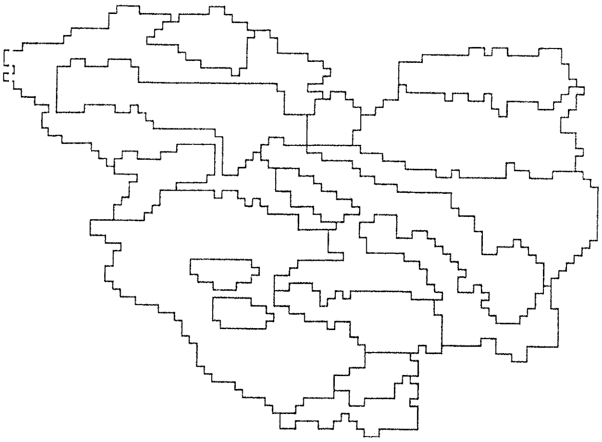

Fig. 15 A vector map from linear quadtree.

REFERENCE

Badji, N. 1990, An object-oriented GIS based on linear quadtree, Proceedings of EGIS'90.

Ehler at al, 1989, Integration of Remote Sensing and Geographical Information Systems: A Necessary Evolution, Photogrammetry Engineering and Remote Sensing, No. 11, 1989

Egenhofer, M. & Frank, A. 1989a, Object-oriented engineering considerations for future GIS, Symposium, Baltimore, MD, USA.

Egenhofer, M. & Frank, A. 1989b, Object-oriented modeling in GIS: Inheritance & Propagation, Proceedings of Auto-carto 9.

Fritsch, D. 1990, Towards three-dimensional data structures in geographical information systems, Proceedings of EGIS'90.

Gong, Jianya. 1990a, Object-oriented models for thematic data management in GIS, Proceedings of EGIS'90.

Gong, Jianya. 1990b, A DTM software based on personal computer, Proceedings of Sym. of Comm. III of ISPRS, Wuhan, China.

Gong, Jianya. 1991, Unified data structure and integrated spatial database in GIS. Doctorate dissertation, Wuhan Technical University of Surveying and Mapping.

Jackie, S.1988, Integration of a GIS and a DTM, 16th ISPRS Congress Comm. III. B3.

Mark, D., Lauzon, J. & Cebrian, J. 1989, A review of quadtree-based strategies for interfacing coverage data with digital elevation models in grid form, INT. J. Geographical Information Systems, 1989 Vol. 3, No. 1.

Molenaar, M. 1989, Single valued vector maps- a concept in geographic information systems, INT. J. Geographical Information Systems, 1989, No.1.

Molenaar, M. 1990, A formal data structure for 3-D vector maps. Proceedings of EGIS'90.

Molenaar, M. & Fritsch, D. 1990, Combined data structures for vector and raster representations in geographical information systems, Symposium of Comm. III of ISPRS, Wuhan, China.

Shaffer, C. etc. 1990, QUILT: a geographic information system based on quadtrees. INT. J. geographical information systems, 1990, Vol. 4, No. 2.

Sonne, B. & Zillien, B. 1990, Hybrid geo-information systems used for environmental protection, Proceedings of EGIS'90.

Yong, Hongguang 1990, Geometry and Topology of Quadtree Decomposition for GIS: Transformation from Raster to Vector Data, EGIS'90, Amsterdam, The Netherlands.