A MIX GLOBAL DATA STRUCTURE BASED ON QTM AND VORONOI

Xuesheng Zhao [a] Jun Chen [b] Zhilin Li [c]

[a] China University of Mining and Technology (Beijing),D11 Xueyuan Road, Beijing, China, 100083,E-mail: zxs@cumtb.edu.cn
[b] National Geometrics Center of China,No.1 Baishengcun, Zizhuyuan, Beijing, China, 10004,E-mail: chenjun@nsdi.gov.cn
[c] Department of Land Surveying and Geo-Informatics The Hong Kong Polytechnic University, Kowloon, Hong Kong
Tel: (+852) 2766 5960, Fax: (+852) 2330 2994; E-mail: lsllz@polyu.edu.hk

Commission IV WG IV/8

KEY WORDS□GIS, Updating, Modeling, Dynammic, Triangulation

ABSTRACT:

To efficiently store, retrieve and update the global spatial data, a hierarchical and dynamic data model on a global scale is urgently needed. The QTM (Quaternary Triangular Mesh) as a hierarchical quadtree data structure on sphere is one of the most efficient methods for managing the global spatial data in many applications. But QTM structure is based on fields instead of objects. So it is efficient in multi-resolution manipulation, but difficult in local data update frequency. To efficiently manipulate multi-resolution and to update data dynamically, a mix global data structure-----Variable Tree Data Structure (VTDS) is designed. There are two different types of 'node's in VTDS, one is 'O_Node' (i.e. object-node), the other is 'I_Node' (i.e. index-node). At every level (except the root one), all 'O_Node's only consist of spatial objects represented by QTM address codes which is efficient in multi-resolutions manipulation. However, 'I_Node's may consist of child 'O_Node's and 'I_Nnode's which include indexing information by which the interested objects within a location scope can be retrieved easily. Meanwhile, the Voronoi diagram based on QTM of related objects at a given level will be dynamically generated to preserve adjacency relationships, which are fundamental to perform queries and updates in local addition or deletion of individual objects. Multi-resolutions manipulation and dynamic update of spherical objects in VTDS are given in details in this paper.

1. INTRODUCTION

In recent years, the QTM (Quaternary Triangular Mesh), which is a efficient spherical hierarchical quadtree data structure for managing the global spatial data, has attracted much attention from the GIS community and has been applied to some areas, such as rendering and managing spherical data [Fekete 1990], spatial data hierarchical index [Goodchild *et al*. 1991], environment monitoring [White *et al*. 1992], positional uncertainty in spatial databases [Dutton 1999b], digital map generalization [Dutton 1999a], global navigation [Lee and Samet 2000], and Continuous indexing of hierarchical subdivisions of the globe [Bartholdi and Goldsman 2001]. QTM partition is a tessellation of the Earth's surface with non-overlapping triangles. It is a field-based data model (or space-primary by Lee et al., 2000) instead of object-based (or feature-primary by Lee et al., 2000). It has the property of obvious hierarchical structure, which is efficient in multi-resolution manipulation. However, such a data model describes hierarchical space using a collection of basic units (e.g. triangle) instead of spatial objects themselves. In this way, if a single spatial object is changed, the entire hierarchical structure may have to be re-organized [Pang and Shi 1998]. The process of re-organization obstructs the maintenance of spatial relations among the changed spatial objects in the hierarchy. Therefore, QTM is not suitable for spatial processes of frequent data updating locally. As a result, a data model with the good properties of both field-based and object-based is very desirable for global data sets, if feasible.

Indeed, in this paper, a Global Hierarchical and Dynamic Data Model (GHDDM) based both on fields and objects will be presented. In this model, spherical surface is partitioned with Quaternary Triangular Meshes and spatial objects are represented by triangular codes, which have both hierarchical and positional properties.

Following this introduction is section for a critical examination of existing global data models (GDMs). In section 3, partition method of QTM on spherical surface and its code labeling will be introduced. Section 4 will present the concept of data model and the representation of spherical objects (points, arcs, and curve surfaces). In section 5, the *VaribleTree Data Structure* (VTDS) based on QTM and Voronoi is described in details. Multi-scale manipulation is discussed in Section 6 and dynamic manipulation of for local updating is discussed in section 7. At last, the conclusions are made in Section 8.

2. GLOBAL DATA MODELS: A CRITICAL EXAMINATION AND A NEW STRATEGY

In the recent years, Global Data Models (GDMs) have been has been a hot topic due to the development of Digital Earth (DE). These models could be either field-based or object-based [Pang & Shi 1998]. The field-based is also called space-primary and the object-based is also called feature-primary (Lee et al., 2000).

In an object-based data model, spatial object representations are in vector mode. The basis for calculating the geographical relationships between surface objects is usually longitude/latitude coordinates, which bring about several problems to be resolved [Lukatela 1987, Gold 1997], such as

- causing coordinates discontinuities of 180 longitude,
- being forced to use conventional ellipsoidal trigonometry involving the transcendental functions which can affect performance greatly, and
- (the most important) formulae being unstable at high latitude

The inadequacies of GDMs based on the latitude/longitude have led a number of researchers to explore alternative approaches. Lukatela [1987] sets up a digital geo-positioning model, in which direction cosines was used instead of latitude/longitude in order to have a seamless data structure. Once the direction cosines for an object are computed, there is no need of using transcendental functions in subsequent spatial manipulations. Instead, calculations are carried out using a vector algebra requiring only floating addition, subtraction, multiplication, division, and an occasional square root. Such calculations are not only very fast but also very stable everywhere on the Earth's surface. On this basis, Gold and Mostafavi [2000] consider to take advantage of the dynamic Voronoi data structure for the addition and deletion of points and line segments using a small set of purely local operations, and for the preservation of the adjacency relationships between objects which are fundamental to perform queries and updates.

But in all object-based GDMs, the hierarchy of space is created by grouping and organizing spatial objects according to some pre-defined relations. In this case, changes are referred to spatial objects themselves and hierarchy of spatial object is maintained using explicitly defined relations among spatial objects instead of recursive decomposition of space. When a spatial process results in changes to spatial objects at one particular level, these changes cannot be propagated to its adjacent levels [Pang and Shi 1998]. Hence, these object-based GDMs are difficult to manage large volume of global data and to manipulate multi-resolution data efficiently.

In a field-based data model, object representations are in cells (i.e. grids or raster) and spherical surface are tessellated as a series of packets or cells. The familiar latitude/longitude graticule (or cell) is the most common basis for GDMs in use today. In the case of GDMs for storing field data, lines spaced at regular latitude and longitude increment form the boundaries of area cells. GDMs based on the latitude/longitude graticule have numerous practical advantages and have been used to develop sound survey sampling designs on the Earth's surface, such as in environment monitoring [Olsen *et al.* 1998] and climate modelling [Thuburn 1997]. These GDMs based on spherical grids would not only allow the same structure to be used over a wide range of spatial resolutions and efficiently load only needed segments [Faust *et al*, 2000], but also allow the presentation of data at multiple levels and any arbitrary resolution and offer several major advantages, such as being unique and domain-independent, appropriately indexed or linearized grids express spherical surface location in a single string, preserving geometrical integrity both locally and globally, and making resolution explicit in the length of the string [Goodchild and Yang, 1992]. GDMs with hierarchical grids properties have been adopted in many contexts, including the quadtree indexes used in spatial database [Samet, 1989], global environment monitoring [White *et al*, 1992], map generalization [Dutton 1999a, 1999b], and dynamic navigation [Lee and Samet 2000]. But their major disadvantages are that (1) they are familiar to a large community, and (b) lack of an intuitive relationship between pairs of codes and proximity [Goodchild 2000]. These properties may be not very important for virtualisation purposes, but would be problematic in maintenance an object in this hierarchical data structure may exist in the nodes of different branches in the tree structure. If an object is removed a little bit (or deletion or insertion), the whole data structure may be changed completely. Therefore, these field-based GDMs are not good for frequent local updating and the consistent topological structure maintenance dynamically [Pang and Shi 1998].

In order to efficiently store, retrieve and analyse spatial data on a global scale, alternatives to current GIS data models are urgently needed [Goodchild and Yang 1992]. The new data model must be

- seamless on a global scale
- efficient for dynamical updating
- capable of facilitating hierarchical representation and
- able to retain the topology of the earth's surface either in the data model itself, or in the internal coordinate system, which allows local modifications and queries.

Our approach starts with a Quandary Tessellation Mesh (QTM) based on the inscribed octahedron, which is used to set up the concept data model of spatial objects. Then, a new hierarchical data structure is constructed by two types of nodes, one is '*O_Node*' for a powerful hierarchical organizing of multi-resolutions data and the other is '*I_Node*' for index mechanism to retrieve local data in a limited viewing window efficiently. Meanwhile, the Voronoi diagram based on triangular-grids between objects at a given level will be dynamically generalized to preserve adjacency relationships, which are fundamental to perform queries and updates in local addition or deletion of individual objects.

3. CONCEPTUAL DATA MODEL OF SPHERICAL OBJECTS BASED ON QTM

The details of the tessellation method on sphere surface and its labelling scheme of QTM can be seen in [Dutton 1999]. In this paper, spherical objects will be represented by QTM codes and their identifiers to save computer storage and to facilitate multi-resolution manipulations.

3.1 Point Objects

Digital representation of a point on the sphere is simple: it consists of an identifier and a QTM address code. QTM address codes consist of digits '0', '1', '2', or '3' except the initial one, and each digital can be expressed by 2-bit. If 32 digits are used in a QTM code, locational accuracy reaches sub-millimeter and such a code can be expressed as one 64-bit word. In conceptual data model, address code of a point is not only used to provide a multi-resolution operation of large-volume global data, but also used to provide a numerical solution of metric problem (by transformation between address code and latitude/longitude coordinates [Goodchild and Yang 1992, Dutton 1999b]). Address codes of points have both hierarchical property and location property. Location of point is implicit in address codes, not as explicit as in the other systems in which the coordinates are records explicitly.

3.2 Arc-line Objects

Arc-lines on sphere are represented by an ordered list of triangles traversed by the arc and a list of vertices in a point format described above, shown as table 1. If the application requires frequent manipulations of spatial multi-scale display and overlay, it may be efficient to only use QTM address codes.

L_1: $ID+$

P_1	$a_{11}a_{12}a_{13}\cdots\cdots a_{1n}$
P_2	$a_{21}a_{22}a_{23}\cdots\cdots a_{2n}$
……	……
P_k	$a_{k1}a_{k2}a_{k3}\cdots\cdots a_{kn}$

Tab.1 ***Arc-line***s on sphere are represented by an ordered list of triangles.

Arc-line is one of the most important objects in a global spatial database (like line in the planar database). For example, hydrology, transportation, terrain relief and region boundaries are commonly represented by arc-lines. For arc-line data, only the vertex points are transferred to corresponding triangle address codes in order to saving storage memory. In most cases, these triangles are discontinuous and may need interpolation (to large scales) and generalization (to small scales). A line can be interpolated to achieve higher levels of detail or filtered to achieve lower levels for display or other manipulations. If two consecutive triangles are not neighbours, interpolate their longitude/latitudes and transfer them to triangular address until consecutive pairs of triangles are neighbours or have the same triangular address. If adjacent triangles have the same triangular address, remove the duplicated ones.

3.3 Area (Curve surface) Objects

Two-dimensional object (curve surface) on the sphere is represented by a directional and closed boundary arc-line and encoded triangular cells, which are completely within the region [shown as Figure 5 and Table 2]. When compared to simple boundary arc circular vertex list, this structure makes the evaluation of spatial relationships significantly more efficient. The solution will often be obtained by simple manipulation of QTM address code, instead of the evaluation of boundary geometry. The time consumption of calculation is inversely proportional to the triangle size.

A_1: ID+

Arc-L_k	P_1	$a_{11}a_{12}a_{13}\cdots\cdots a_{1n}$
	P_2	$a_{21}a_{22}a_{23}\cdots\cdots a_{2n}$
	……	……
	P_k	$a_{k1}a_{k2}a_{k3}\cdots\cdots a_{kn}$
Encoded triangles-N_j	N_1	$b_{11}b_{12}b_{13}\cdots\cdots b_{1n}$
	N_2	$b_{21}b_{22}b_{23}\cdots\cdots b_{2n}$
	……	……
	N_j	$b_{j1}b_{j2}b_{j3}\cdots\cdots b_{jn}$

Tab.2 ***Area*** (Curve surface) is represented by a directed circular boundary arc-line and an encoded aggregate list of triangle cells.

This representation of a two-dimensional object is a combination of the traditional vector representation and those schemes based on regular planar tessellations. It is of high resolution and precision as vector representation and is efficient in relational evaluation as raster. In addition, it does not violate the true spherical nature of the data domain. For instance, if *[A]* is a region, then *NOT[A]* is an infinite, numerically ill-defined region in a planar. By contract, on any spherical surface, *NOT[A]* is the simple finite complement.

4. VARIBLE-TREE DATA STRUCTURE

4.1 The concept of variable tree structure

The dynamic manipulation (or updating) of spatial data is based on objects, and multi-resolution management of the global spatial data is based on fields instead of objects. To accommodate these two requirements, a mixed global data structure -- *Variable Tree Data Structure* -- is designed by connecting the dynamic stability of Voronoi data structure with the hierarchy of QTM. In this data structure, the spherical surface is tessellated with QTM and spatial objects are represented by triangular codes, which have both hierarchical and positional properties.

To operate data dynamically and efficiently, VTDS is reconstructed with two types of new nodes which are different with the traditional quadtree data structure:

- One is '*O-Node*' (i.e. object-node) for a standard QTM hierarchical organizing of multi-resolutions data. The sub-node of this type consists of '*leaf-node*' and '*branch-node*' which is as the same as the traditional quadtree.
- Another is '*I_Node*'(i.e. index-node) which expresses an index mechanism for retrieving the required data level and local objects in a limited viewing window efficiently. The sub-node of this type consists of '*O-Node*' and '*I-Node*' in next level.

Meanwhile, the Voronoi diagram between objects at a given level will be dynamically generated to preserve adjacency relationships which are fundamental to perform queries and updates in local addition or deletion of individual objects. The basic principle of VTDS is illustrated in Figure 1. The representation of *O_Nodes* is almost same as that by the traditional quadtree, so in this section, only the different types of *I_Node*s and their representations in VTDS will be discussed in details.

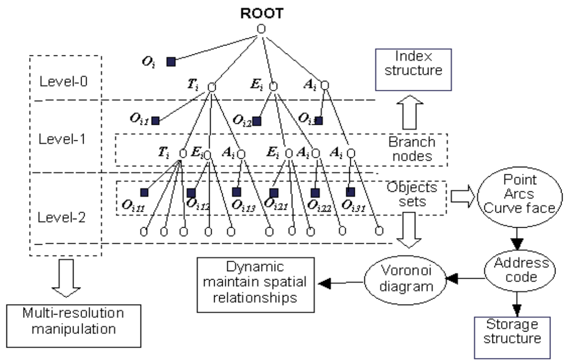

Fig.1 Basic principle of ***VaribleTree Data Structure (VTDS)***

4.2 Initial Nodes

From the tree root, the VTDS starts with as a forest of initial three types of *I_Nodes* and one type of *O_Node* according to their different locations. They are defined as follows (shown as Figure 2):

- *In-triangle (T)*: The object is completely within an octant triangle (including initial triangles T_0~T_7, totally 8 *I_Nodes*);
- *Edge-neighbour (E):* The object only covers two *edge-neighbour* octant triangles (including E_{01}, E_{12}, E_{23}, E_{30}; E_{45}, E_{56}, E_{67}, E_{74}; E_{04}, E_{15}, E_{26}, E_{37}, totally 12);
- *Angle-neighbours (A)*: The object covers two or more *angle-neighbour* octant triangles which have one common vertex (including: A_{0123}, A_{4567}, A_{0374}, A_{1045}, A_{3267}, A_{2156}, totally 6);
- *NO-neighbour* (Objects): An object covers two or more no-neighbour octant triangles, these are *O_Nodes* and the pointers, which point to these objects, are stored in the nodes (including: O_{06}, O_{17}, O_{24}, O_{35}).

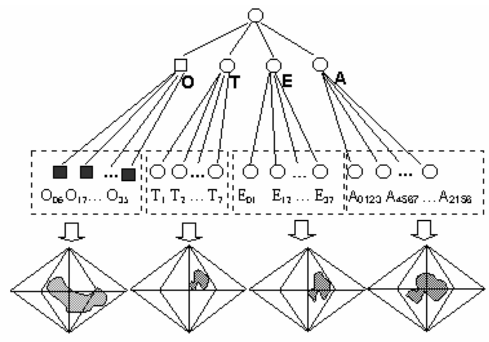

Fig.2 Categories and representations of ***initial*** *Nodes*

4.3 In Triangular Node (T)

The spherical object within a triangle can be classified into 4 categories at next level according to their locations. They are defined as follows (shown as Figure 3):

- *In Sub-triangle*: The object is completely within a sub-triangle (expressed as S_i (i=0,1,2,3) , totally 4).
- *Edge-neighbour*: The object covers two *edge-neighbour* sub-triangles (expressed as E_{0i} (i=1,2,3), totally 3).
- *Angle-neighbours*: The object covers two or more *angle-neighbour* sub-triangles which have one common vertex (expressed as A_{012}, A_{023}, A_{013}, totally 3).
- *NO-neighbour* (Objects): The object traverses the four sub-triangles. It is *O_Node* and the pointer which point to the object is stored in this node. It is expressed as O_i (i=1,2,…,n).

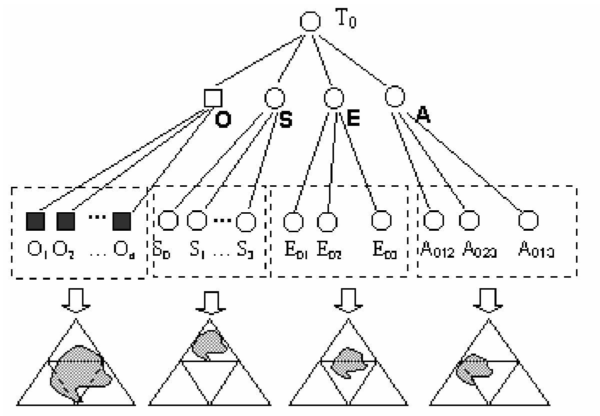

Fig.3 Sub-*I_Nodes* categories from in-triangle and their representations

4.4 Edge-neighbour Node (E)

Spherical objects, which cover two edge-neighbour sub-triangles, can be classified into 3 categories according to their locations as follows (Figure 4):

- *Edge neighbour:* The object covers only two *edge-neighbour* sub-triangles (expressed as E_{22}, E_{33}, totally 2)
- *Angle-neighbours*: The object covers two or more *angle-neighbour* sub-triangles which have one common vertex (expressed as A_{23}, totally 1)
- *NO-neighbour* (Objects): The object covers the *no-neighbour* sub-triangles, it is *O_Node* which store a pointer pointing to this object (expressed as O_1).

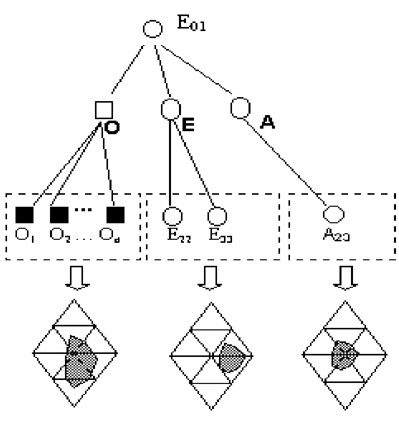

Fig.4 Sub-*I_Nodes* categories from Edge-neighbour node and representations

4.5 Angle-neighbour Triangles Node (A)

Spherical objects which covers two or more *angle-neighbour* sub-triangles with one common vertex can be classified into 2 categories according to their locations as follows (shown as Figure 5):

- *Angle-neighbours:* The object covers two or more *angle-neighbour* sub-triangles with one common vertex (expressed as A^{-0}, totally 1).
- *NO-neighbour (Objects):* The object covers *no-neighbour* sub-triangles, it is *O_Node* which stores a pointer pointing to this object (expressed as O^0).

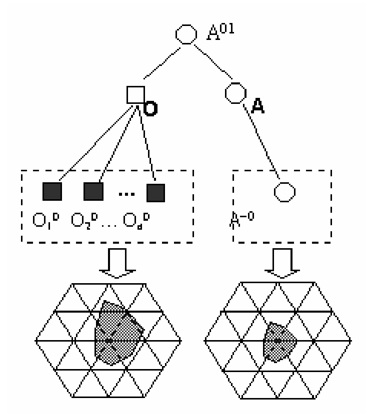

Fig.5 Sub-*I_Ndes* categories from Angle-neighbour node and representations

5 DYNAMIC MANIPULATION OF LOCAL SPATIAL DATA IN VDTS

A change in local area may result in the reconstruction of topological relationship in a larger area [Gold 1992]. This process obstructs the real-time updating of the local spatial data and limits applications greatly. In this section, we utilize the property of dynamic stability of Voronoi data structure to update the individual objects while this process only influences the neighbour objects and keeps the relationship structure of other objects stability or changeless.

5.1 Insertion and Deletion of Spatial Objects

Insertion and deletion of spatial objects are the two main dynamic manipulations in local data updating. In this section, we discuss how to insert an object to its right node location in VTDS in details.

Each object can be expressed by QTM address codes in VTDS. Let the triangular address codes of object ***U*** be represented by

$$u = u_{n_0}^{0} u_{n_1}^{1} \ldots\ldots u_{n_k}^{k}$$

where k is the levels of VTDS data structure; n_k is the total number of address codes in object ***U*** in level k (<=8×4^k). Digitals in U^k (*k=1,2,…*) include '0', '1', '2' and '3' and U^0 include digitals '0'~'7' in primitive level 0. We use code $\boldsymbol{t^i}$, $\boldsymbol{e^i}$, and $\boldsymbol{a^i}$ to represent *I_Nodes* in level *i* whose object is completely within one triangle, covering two edge neighbour triangles, and covering more than two angle neighbour triangles which have one common vertex respectively. Code $\boldsymbol{E^i}$ represents *I_Nodes* in level *i* whose object traverses two edge neighbour triangles in different octant triangles. Code $\boldsymbol{A^i}$ represents *I_Nodes* in level *i* whose object traverses more than two angle neighbour triangles in different octant triangles in which have one common vertex.

1). To search $u_{n_0}^0$ in level 0 (type 1):

- If $\{0,6\}.OR.\{1,7\}.OR.\{2,4\}.OR.\{3,5\} \in \{u_{n_0}^0 \mid n_0=0,1,\ldots,7\}$, Object[$u$]→*level 0*, then stop recursive search.
- If $n_0=1$ *I_Node* T_{n0} is added if it is not existed and process $u_{n_1}^1$ according to type 2.
- If $n_0=2.AND.(u_1^0, u_2^0) \in \{(2,6),(3,7),(0,4),(1,5)\}$, *I_Node* E_{U1u2} is added if it doesn't exist, process $u_{n_1}^1$ according to type 3
- If $n_0=2.AND.(u_1^0, u_2^0) \in \{(0,1),(1,2),(2,3),(3,0),(4,5),(5,6),(6,7),(7,4)\}$, *I_Node* A_{U1u2} is added if it is not existed and process $u_{n_1}^1$ according to type 6.
- Otherwise, *I_Node* $A_{U1u2U3U4}$ is added if it doesn't exist, process $u_{n_1}^1$ according to type 5.

2). To search *I_Node* t_i in which object is within one triangle level i $(i<k)$ (type 2).

- If $n_i = 4$, the search finishes, and object [U] →level i.
- If $n_j = 1.AND.\{u_1^i\} \in \{0,1,2,3\}$, *I_Node* t_{u1} is added if it doesn't exist, process $u_{n_{i+1}}^{i+1}$ according to type 2.
- If $n_j=2.AND.(u_1^i, u_2^i) = (0,1)$, *I_Node* e_{u1u2} is added if it doesn't exist, process $u_{n_{i+1}}^{i+1}$ according to type 3.
- If $n_j=2.AND.(u_1^i, u_2^i) \in \{(0,2),(0,3)\}$, *I_Node* e_{u1u2} is added if it doesn't exist, process $u_{n_{i+1}}^{i+1}$ according to type 4.
- Otherwise, *I_Node* a_{u1u2u3} is added if it doesn't exist, process $u_{n_{i+1}}^{i+1}$ according to type 5

3). To search *I_Node* e_{u1u2} *(or* E_{u1u2}*)* in which object traverses two edge-neighbour triangles which neighbour orientation is '*Top and Down*' in level i $(i<k)$ (type 3):

- If digital '1' $\in \{u_{n_i}^i \mid n_i=0,1,\ldots,\}$, the search finishes, and object[U] →level i.
- If $(u_1^i, u_2^i) \subset \{(2,2), (3,3)\}$, *I_Node* e_{u1u2} is added if it doesn't exist, process $u_{n_{i+1}}^{i+1}$ according to type 3
- Otherwise, *I_Node* $a_{u1u2u3u4u5u6}$ is added if it doesn't exist, process $u_{n_{i+1}}^{i+1}$ according to type 5.

4). To search *I_Node* e_{u1u2} in which object traverses two edge-neighbour triangles in one octant which neighbour orientation is '*Left-Right*' in level i $(i<k)$ (type 4):

- If $u_1^i = 2.OR.\, u_j^i = 3$, the search finishes, and object [U] →level i.
- If $n_j=2.AND.(u_1^i, u_2^i) \subset \{(1,2), (3,1)\}$, *I_Node* e_{u1u2} is added if it doesn't exist, process $u_{n_{i+1}}^{i+1}$ according to type 4..
- Otherwise, *I_Node* $a_{u1u2u3u4u5u6}$ is added if it doesn't exist, process $u_{n_{i+1}}^{i+1}$ according to type 5.

5). To search *I_Node* $a_{u1u2u3u4u5u6}$ *(or* $A_{u1u2u3u4}$*)* in which object traverses two more than two angle-neighbour triangles that have one common vertex in *level i* $(i<k)$ (type 5).

- If '0' $\not\subset \{u_{n_i}^i \mid n_i=0,1,\ldots,j\}$, *I_Node* a^i is added if it doesn't exist, process $u_{n_{i+1}}^{i+1}$ according type 5.
- Otherwise, the search finishes, and object [U] →level i.

6). To search *I_Node* E_{u1u2} in which object traverses two edge-neighbour triangles in two different octant and its orientation is '*Left-Right*' in level i $(i<k)$ (type 6):

- If $u_1^i = 2.OR.\, u_j^i = 3$, the search finishes, and object [U] →level i
- If $n_j=2.AND.(u_1^i, u_2^i) \subset \{(1,1), (3,2)\}$, *I_Node* E_{u1u2} is added if it doesn't exist, process $u_{n_{i+1}}^{i+1}$ according to type 6.
- Otherwise, *I_Node* $A_{u1u2u3u4u5u6}$ is added if it doesn't exist, process $u_{n_{i+1}}^{i+1}$ according to type 5.

The deletion operation is almost as same as insertion. The only difference is that the searching orientation is reversed.

5.2 Adjacency Relationship Maintenance in Dynamic Manipulation

In local updating, deletion and insertion of objects results in a change in spatial relationship with only adjacent objects and indeed the topological relationships of other objects remain unchanged (shown as Figure 6). This is one of the excellent properties of Voronoi diagram [Gold 1992]. In VTDS, QTM-based method for the computation of a spherical Voronoi diagram can be seen in our former work [Zhao *et al*. 2002]. The whole data of one object is in one node and it is very easy to calculate the adjacency relationships of the changing objects and their neighbours.

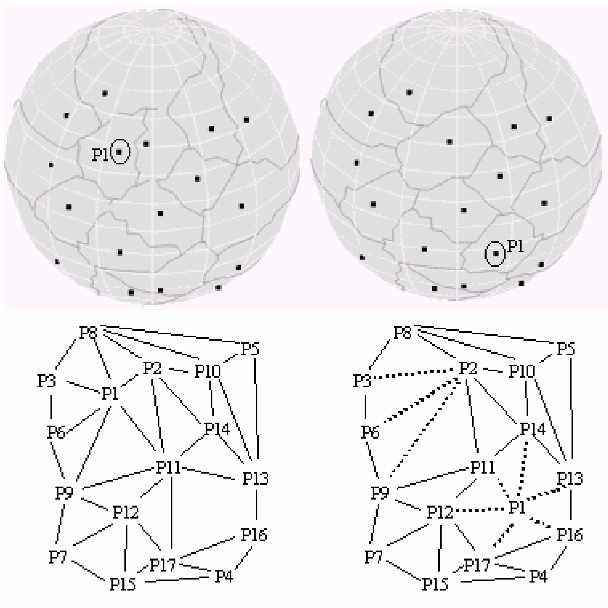

Fig.6 Deletion and insertion of objects and their adjacent relationships changes..

6 DISSCUSION AND CONCLUSIONS

In this GHDDM, the conceptual data model of spherical objects and their representation based on QTM address codes are presented at first. It offers several advantages, such as being unique and domain independent, appropriately indexed or linearized grids express spherical surface location in a single string, preserving geometrical integrity both locally and globally, and making resolution explicit in the length of the string.

One important contribution in our approach is that integrate the advantages of field-based and object-based data models to construct the *VaribleTree* Data Structure (VTDS) by two types of new nodes, one is *O_Node* for a powerful hierarchical organizing of multi-resolutions data and another is *I_Node* for index mechanism to retrieve local data in a limited viewing window efficiently. Furthermore, the VTDS provides a geo-spatial data indexing mechanism with which one can navigate continuously from global overviews to high-resolution local views.

Another contribution in our approach is that the Voronoi diagram is embedded in VTDS. Voronoi diagram between objects at a given level can be dynamic generalized by recursive dilation of spherical triangles based on QTM address codes. The adjacency relationships from Voronoi-diagram, which is fundamental to perform queries and updates, allow the addition or deletion of individual objects using a small set of purely local operation.

ACKNOWLEGEMENT

The work described in this paper was substantially supported by an Outstanding Youth Award from the Natural Science Foundation of China (under grant No.40025101) and by CERG project from the Research Grants Council of the Hong Kong Special Administrative Region (Project No. PolyU 5048/98E).

REFERENCE

Bartholdi, J. III., and Goldsman, P., 2001, Continuous indexing of hierarchical subdivisions of the globe. *International Journal of Geographical Information Science,* Vol.15, No.6, 489-522.

Dutton, G., 1999a, Scale, sinuosity, and point selection in digital line generalization. *Cartography and Geographic Information Science*, Vol.26, No.1, 33-53.

Dutton, G., 1999b, A Hierarchical Coordinate System for Geoprocessing and Cartography. *Lecture Notes in Earth Sciences,* Springer-Verlag, Berlin, 230pp.

Fekete, G., 1990, Rendering and managing spherical data with sphere quadtree. *Proceedings of Visualization'90*, October, San Francisco, California (IEEE Computer Society Press), pp176-186.

Gold, C.M., 1992, The meaning of '*Neighbor*'. *Theories and Methods of Spatio-Temporal Reasoning in Geographic Space*, Lecture Notes in Computing Science 639, Springer-Verlag, Berlin, pp220-235.

Gold, C.M., 1997, The global GIS. *Proceeding of the 1st International Workshop on Dynamic and Multi-Dimension GIS.* August, The Hong Polytechnic University, Hong-Kong, China, pp 80-91.

Gold, C.M., and MOSTAFAVI, M., 2000, Towards the global GIS. *ISPRS Journal of Photogrammetry and Remote Sensing*, Vol.55, No.3, 150-163.

Goodchild, M.F., and YANG, S.R., 1992, A hierarchical data structure for global geographic information systems. *Computer Vision and Geographic Image Processing*, Vol.54, No.1, 31-44.

Goodchild, M.F., YANG, S.R. and DUTTON, G., 1991, Spatial Data Representation and Basic Operations for a Triangular Hierarchical Data Structure. *NCGIA Report 91-8*, National Center for Geographic Information Analysis, University of California, Santa Barbara, California, 14pp. (available at http://www.ncgia.ucsb.edu, last accessed at 10 August 1999).

Goldchild, M.F., 2000, Discrete Global Grids For Digital Earth. *Proceeding of the 1st International Conference On Discrete Global Grids*, March, Santa Barbara, California, available at http://www.ncgia.ucsb.edu/globalgrids/papers, last accessed at 15 November 2000.

JoneS, C., 1997, Geographical Information Systems and Computer Cartography. *Longman Singapore Publishers Ltd.*, 319pp.

Lee, M., and Samet, H., 2000, Navigating through triangle meshes implemented as linear quadtree. *ACM Transactions on Graphics*, Vol.19, No.2, 79-121.

Lee, YC., Li, Z.L. and Li, Y.L., 2000, Taxonomy of space tessellation. *ISPRS Journal of Photogrammetry and Remote Sensing,* No.55, 139-149.

LukatelA, H., 1987, Hipparchus geopositioning model: An overview. *Proceedings of the Eighth International Symposium on Computer-Assisted Cartography*, 29 March – 03 April, Baltimore, Maryland, pp. 87-96.

Olsen, A., StevenS, D., and White, D., 1998, Application of global grids in environmental sampling. *Computing Science and Statistics,* No.30, 279-284.

Otoo, E., and Zhu, H., 1993, Indexing on spherical surfaces using semi-quadcodes. *Advances in Spatial Databases 3rd International Symposium, SSD`93*, June, Singapore, Lecture Notes in Computer Science 692, Springer, Singapore, pp.509-529.

Pang, M. and Shi, W., 1998, Modeling Hierarchical Structure of Spatial Processes Using Voronoi Spatial Model. *8th International Symposium on Spatial Data Handling, SDH'98*, July, Vancouver, Canada, pp.34-43.

Thuburn J., 1997, A PV-based shallow-water model on a hexagonal-icosahedral grid. *Monthly Weather Review*, No.125, 2328-2347.

White, D., Kimmerling, J. and Overton, W.S. 1992, Cartographic and geometric components of a global sampling design for environment monitoring. *Cartography & Geographical Information Systems*, Vol.19, No.1, 5-22.

Zhao, X.S., Chen, J. and Li, Z.L., 2002, A QTM-based algorithm for the generation of Voronoi diagram on sphere. *Advances in Spatial Data Handling*. Published by Springer-Verlag in Beilin. pp.269-284.