A Kind Index Structure based on VGI information Combined Query

Lingli ZHAO[1, a] ,Shuai LIU[1], Junsheng LI[1]

[1] School of Engineering, Honghe University, Mengzi, 661100, China

[a]email: zll@126.com

Keywords: Hiberarchy index tree; R-trees; classified index; joint inquiry; urbanization data

Abstract. The paper proposes a kind of data index structure - the hiberarchy index tree to implement joint inquiry, this structure can expand the search range and has the ability to quickly query. The experiments confirmed that the proposed the hiberarchy index tree can create spatial index, implement a direct link between MBR based index and non-spatial feature, and can quickly query multiple data sources. The experiment shows that the hiberarchy index tree is much validated and something useful is obtained.

Introduction

Web 2.0 technologies enable users of social media to make contributions or to communicate with each other. Among the various types of information contributed and shared by users on social media, the geographic one is called Volunteered Geographic Information (VGI)[1]. VGI has demonstrated potential to enrich the user experience of products which utilizes geo-information, including enhancing credibility[2], increasing spatial accuracy [3] and the collection and distribution of VGI through mobile devices [4]. VGI has been used in research on tourism, disaster and crisis management, transportation, etc. The most common providers of VGI are Flickr, OpenStreetMap, Twitter, Facebook, YouTube, Wikimapia, Foursquare, etc. The amount of valuable data in VGI and different structures grew and technical progress made it possible to link these different systems and different structures, the wish to exchange and share these data arose and became more and more important. But the combination of data done on different levels that we subsume under the term data integration [1] requires resolving heterogeneities, which still poses research questions [2]. Database people have dealt extensively with architecture alternatives [3].

Much research has been and is still done in order to develop and improve them, for example in the areas of feature matching [4,5,6], generalization [7,8,9] and semantic data integration[10]. For GIS (geographic information system) professionals and users, the urbanization has brought forward new geospatial challenges, and it forces the construction of urbanization in geospatial establishment.

VGI information is a kind of multi-source data absolutely, which contains spatial data and tagging data. There are various kinds of spatial data, some are image raster data, and the others are vector digital data. Tagging information data also have different forms, some are digital form, and others are images. The variety of these data demands that the data integration system could read all types of data, frame uniform standard to classify these data, establish the data relationship among them and integrate usable data resource.

Spatial data and tagging information data combined query is one of the most difficult tasks in data integration. At present, there are two ways, first, spatial data and tagging data stored in the same database, create a data link between the two data to achieve a unified united inquiry, such as: ESRI has introduced universal object-oriented database Geodatabase [4], the second is the use of data object spatial information (for example: coordinate, area, etc.) to establish relation. The former requirement must establish contact when building a database, which means a single query, the query information is limited. VGI data are more heterogeneous data sources, the lack relation of spatial data and tagging data directly, thus linking the above two methods should not be used to establish contact. This is an urgent need to develop a data storage index model to the establishment of the spatial data and tagging data links between them.

There are a lot of spatial data indexing research, such as the R tree family index [11,12,13]. R family tree algorithms are a hierarchical data structure, dynamic index algorithm, using minimum bounding rectangle (MBR) to approximate the complex spatial objects, without an index to predict the scope of the study area, this structure is fit for the characteristics of urbanization spatial data, so the use of MBR based index data storage space.

The paper proposes a kind of data index structure - the hiberarchy index tree to implement joint inquiry, this structure can expand the search range and has the ability to quickly query. The experiments confirmed that the proposed the hiberarchy index tree can create spatial index, implement a direct link between MBR based index and tagging feature, and can quickly query multiple data sources.

THE Hiberarchy Index Tree Structure

The proposed data index structure model is a regional division based on hiberarchy index tree by recording the minimum range for each area rectangle MBR to create a region name and the spatial MBR based index direct link between (Figure 1 left a blue double dashed arrow). Urban thematic data is established by the classified index, and data list range is one of the regional classified indexes, so it established a classified index and the links between MBR based index (upper right in Figure 1 red double dashed arrows). The establishment of spatial data indexing and thematic data offers direct contact for the data integration, and provides the necessary conditions for the joint query for spatial data and thematic data Figure 1 is the level of the index tree and the MBR based index and the thematic data associated diagram.

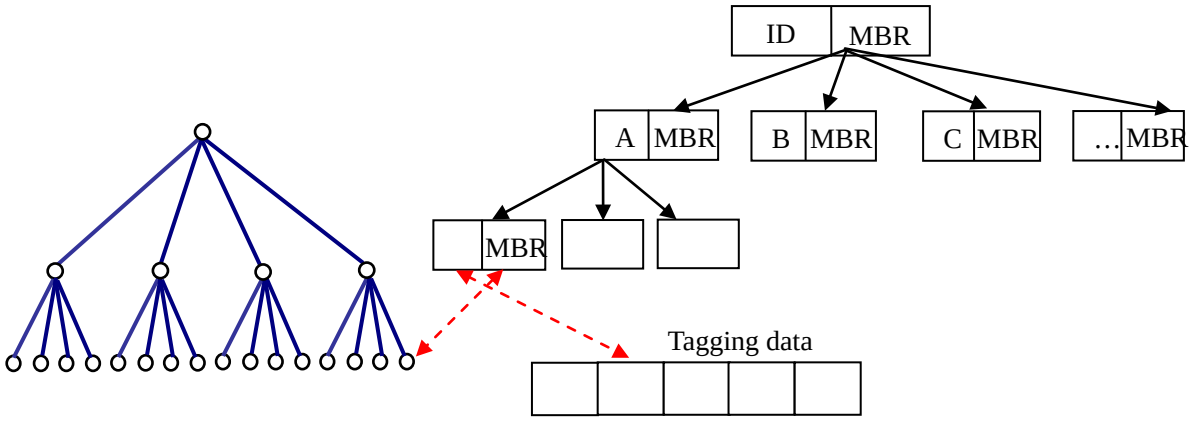

Fig.1. The Flow Chart of Building Relationships among Hiberarchy Index Tree, MBR based index and tagging data

Definition: hiberarchy Index tree is a kind of index structure with regional name and regional number as the main content, and the basic characteristics of a tree. Let M to be the maximum number of child nodes, a non-empty index tree has the following characteristics:

a). Root rode is the highest level and the largest range of areas; the region of the upper node is composed of all nodes from the lower region.

b). Each node consists of four parts: region number; minimum range of the rectangle (MBR), table pointer of a child node, the number of child nodes.

c). The number of each node is m, and satisfied: $0 \leq m \leq M$.

d). Child nodes in the table arranged in ascending number.

According to the definition of hiberarchy index tree, the range of the tree root area is largest, while the region of child nodes is subset of the upper node region.

The hierarchical index tree operations

The most important algorithm is the hierarchical index tree generation and finding relevant information about tree node operation in the definition of the hierarchical index tree. Query algorithms are based on the hierarchical index tree. The following describes the establishment of the

hierarchical index tree algorithm and how to traverse the tree to find information in detail. The flow chart is shown in Figure 2.

(1) The hierarchical index tree establishment:

Function name：CreateTree

Input: regional layers

Output: the established hierarchical index tree

Specific step:

i) generate root nodes of : the hierarchical index tree.

ii) Traverse the region layers, obtaining the number, name, MBR, and tree nodes. When the layers are traversed, skip to step v.

iii) Seek the inserted node's parent node in the hierarchical index tree.

a) initialize and set the current node to the root node.

b) search the hierarchical index tree top-down, and get the order form which is compared by the layer node.

c) traverse the current layer node, compare the number of node to insert and the current node. If the layer number of the current node is not the node number of subset, and then compare continues ones. If the layer number of the current node is not the node number of subset, and the lower node of the current is not empty, set the current node is a parent node. Go to the next node to compare and skip to step (b). If the current node is empty, and then exit loop.

iv) put the node into the hierarchical index tree. Insert the current node to the order form of parent node found in step iii, skip to step ii.

v) Output the established hierarchical index tree

(2) query node information (based on MBR), the flow chart is shown in Figure3.

Function name：FindNodesByMBR

Input: MBR, Spatial relationships (for example: intersect, contain)

Output: Collection of nodes

Specific step:

i) initialize the Hiberarchy index tree queue for traversing, root nodes are put into the queue, and then initialize query results of node collection. and then initialize query results of node collection.

ii) traverse the hiberarchy index tree, looking for nodes which meets certain spatial relationships.

(a) head node of the team (current node) skips out of the queue.

(b) determine the queue, if the queue is empty, go to step iii. Otherwise, all the child nodes of the current node are added to the queue, and between the MBR of the node and the input of MBR to determine the spatial relationship. If so, the current node is added to the result node set, and skip to step (a).

iii) output node sets of query result.

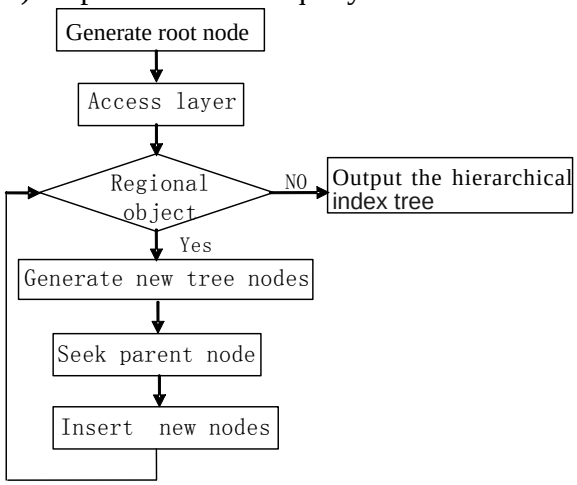

Fig.2. Building Hiberarchy Index Tree

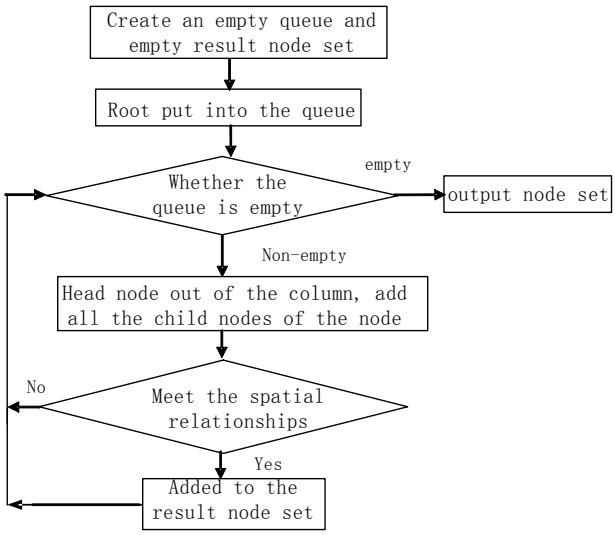

Fig.3. Querying Nodes by MBR

(3) query node information (based on Id): In the hierarchical index tree, query is that all

information of nodes in a subset is input, and the flow chart is shown in Figure 4.

Function name：FindNodesById

Input: Area ID

Output: Collection of nodes

Specific step:

i) initialize query result set of nodes.

ii) look for a set number of areas containing the input node in the hierarchical index tree.

a) Initialize and set the root node to the current node.

b) determine the current node, if it is empty, skip to step iii. Otherwise, compare the current node number and the area number of the input. If it is a subset of the relationship, the current nodes are added to the result node set, and go to the child nodes of the node sequence table to search, and repeat steps (b). If the relationship is equal, go to step iii. If it neither is a subset of the relationship not equal, proceeds to compare sibling nodes, and repeat steps (b).

iii) output query node result sets.

(4) Query MBR (based on ID): in the hierarchical index tree, query input area number directly to MBR information of the parent node, the flowchart is shown in Figure 5, the following specific steps describe the functions and algorithms.

Function name：FindMBRById

Input: Area ID

Output: MBR

Specific step:

i) initialize query results node.

ii) look for direct parent node contains input area number in the hierarchical index tree.

a) initialize and set the root node to the current node.

b) determine the current node, if it is empty, skip to step iii. Otherwise, compare the current node number and the area number of the input. If it is a subset of the relationship, let the query results to be the current nodes, and go to the child nodes of the node sequence table to search, and repeat steps (b). If the relationship is equal, let the query results to be the current nodes, and go to step iii. If it neither is a subset of the relationship not equal, proceeds to compare sibling nodes, and repeat steps (b).

iii) output the node MBR results of the query.

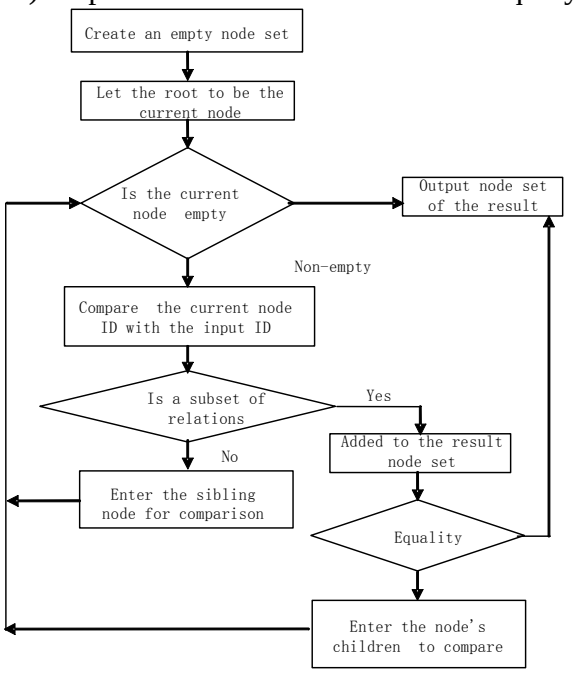

Fig.4. The Flow Chart of Querying Nodes by ID

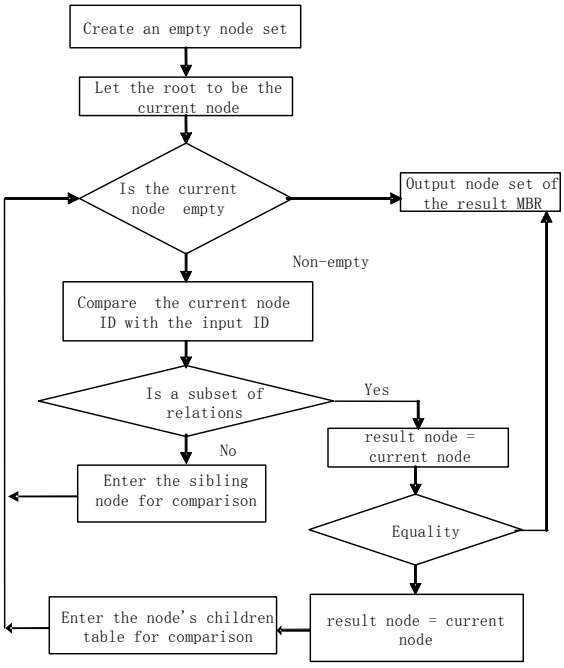

Fig.5. The Flow Chart of Querying MBR by ID

Joint inquiry

There are two kinds of established links between hierarchical index tree and R-tree index

according to the definition and characteristics of hierarchical index tree. First, establish a direct link through the node MBR and R-tree index. Second, establish link by objects ID with R-tree index node ID. In addition, the establishment of hierarchical index tree and attribute data classified index is closely linked. When query spatial data, establish correspondence between MBR in a query to get through the R-tree index or ID by hierarchical index tree, and combined classified index to query relevant information of the attribute data. Establish the logical relationship of spatial data, attribute data, and thematic data by hierarchical index tree, R-tree index and classified index linked. When query attribute data, query in the hierarchical index tree according to the regional ID to obtain available regional MBR, and then can quickly find relevant information of regional spatial data by the R-tree index based on MBR.

Experiment

The experiment takes southern city's urban data as an example in the .NET platform to achieve the combined query function. The attribute data stored in the database, the establishment of a comprehensive and standard on the original data file's metadata, define metadata table, record the original file information. The urban property data can be divided from different angles, the contents of the metadata table to establish classified index for fast retrieval and attribute queries.

The establish hierarchical index tree

Establish hierarchical index tree based on regional data layer, and establish on 18,194 areas by R-tree index in different regions of two experiments. The first group was divided into 508 areas by city, district and town in three different levels. The second group was divided into 2680 areas by city, district, town and villages or streets in four different levels. Experimental results are shown in table 1. Laboratory equipment is a normal laptop. It can be seen from the table 1, the time of establishing hierarchical index tree is not long and the query speed is high even in the case of many objects.

TABLE I. EXPERIMENTS COMPARISON OF HIBERARCHY INDEX TREE

	Regional number	**Regional level**	**Height**	**average construction time**	**average query time**
The first group	508	3	3	0.01s	0.001s
The second group	2680	4	4	0.09s	0.001s

Time and space algorithm time complexity analysis

The node information query algorithms in the hierarchical index tree are very important，the following analysis for time and space complexity of the query algorithm. Let the layers of hierarchical index tree are k, the total number of nodes is n, the maximum number of child nodes is m. When execute the query algorithm in the hierarchical index tree, the visiting number of the function FindNodesByMBR in the hierarchical index tree is $k*n$. Since k is constant, the time complexity of the query is O (n). As the child node ID is orderly, binary search can be performed, the number of function FindNodesById access is $k*log_2m$, so the time complexity of the query is $\boldsymbol{O}(log_2m)$. the number of function *FindMBRById* access is $k*log_2m$, so the time complexity of the query is $\boldsymbol{O}(log_2m)$.Each unit node is set as memory space by a unit, then the maximum auxiliary space required of the query function FindNodesByMBR is m, the space complexity is $\Theta(m)$, the maximum auxiliary space required of the query function *FindNodesById* is k, the space complexity is $\Theta(k)$, and the maximum auxiliary space required of the query function *FindMBRById* is 2, the space complexity is $\Theta(k)$.

TABLE II. SPACE COMPLEXITY FOR PERFORMING QUERY

	Height	**The total number of nodes**	**The maximum number of child nodes**	**FindNodesByMBR maximum auxiliary space required**	**FindNodesById maximum auxiliary space required**	**FindMBRById maximum auxiliary space required**
The first group	3	508	80	80	3	2
The second group	4	2680	80	80	4	2

In summary, the operation efficiency of hiberarchy index tree is high, and auxiliary calculations take up little space. There are three different ways for queries between tagging and spatial data in the hierarchical tree structure. The first is through spatial data query thematic data. The second way is through the thematic data query spatial data and relating information. The third way is through some portion of spatial data and tagging data to achieve comprehensive joint query.

Acknowledgment

This research is supported by National Natural Science Foundation (No. 41301442, 41201418), and Pecuniary aid of Yunnan Province basic research for application （2013fz127）

References

[1] Bömelburg, J., ATKIS-Datenintegration. Das Geoinformations system ATKIS und seine Nutzung in Wirtschaft und Verwaltung. 3. AdV-Symposium ATKIS, Koblenz (pp. 199-204). 1996.
[2] Koch, C., Data Integration against Multiple Evolving Autonomous Schemata, PhD thesis, Technical University (TU) Vienna. 2001.
[3] Dadam, P., Verteilte Datenbanken und Client/Server-Systeme. Grundlagen, Konzepte und Realisierungsformen, Berlin: Springer, 1996.
[4] Gabay, Y. & Doytsher, Y. 1995, Automatic Feature Correction in Merging Line Maps. ACSM/ASPRS Annual Convention & Exposition Technical Papers, (pp. 404-411), Charlotte, North Carolina.
[5] Walter, V. & Fritsch, D. 1997, Matching Strategies for Integration of Spatial Data from Different Sources. International Workshop on Dynamic and Multi-Dimensional GIS, (pp. 215-228), Hong Kong.
[6] Sester, M., Anders, K.-H. & Walter, V. 1998, Linking Objects of Different Spatial Data Sets by Integration and Aggregation. Geoinformatica 2(4), 335-357. Retrieved February 14, 2003 from the Kluwer website:http://www.wkap.nl/article.pdf-193781
[7] Guercke, R., Brenner, C., Sester, M. Data Integration and Generalization for SDI in a Grid Computing Framework. International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, Vol. XXXVII,Beijing, 2008.
[8] Weibel, R. & Jones, C. B. (1998) Computational Perspectives on Map Generalization. Geoinformatica 2(4), 307-314.
[9] Cecconi, A. & Weibel, R. 2001, Map Generalization for On-demand Mapping. GIM International 15(5), 12-15.
[10] Birgit Kieler, Monika Sester, Hannover, Haiqing Wang, Jie Jiang. Semantic Data Integration: Data of Similar and Different Scales. Photogrammetrie, Fernerkundung, Geoinformation 6/2007, S. 447-457
[11] Guttman A.R-Trees:A Dynamic Index Structure for Spatial Searching .Proc.ACM SIGMOD Int.Conf.on Management of Data,1984: 47-57.
[12] Beckmann N.The R*-tree:An Efficient and Robust Access Method for Points and Rectangles[J].Proceedings of the 1990 ACM SIGMOD Conf,1990,6:322-331
[13] T. Sellis, N. Roussopoulos, and C. Faloutsos. The R+ tree: a dynamic index for multi-dimensionnal objects. In Proceedings of the 13th Conference on VLDB , London, England, 1987:507-518