SPATIOTEMPORAL INDEXING MECHANISM BASED ON SNAPSHOT-INCREMENT

L. Lin [a], Y. Z. Cai [a, b], Z. Xu [a]

[a] School of Resource and Environment Science，Wuhan university, Wuhan China 430079, lilin@telecarto.com
[b] School of Resources and Environmental Engineering, Wuhan University Technology, Wuhan, China 430070, yinzhangcai@163.com

KEY WORDS: Version，Increment，Index Mechanism，Spatio-Temporal Model

ABSTRACT:

Spatio-temporal index is one of the key technologies applied to spatio-temporal database engines. Methods for spatio-temporal indexes which consider efficiency of both temporal queries and spatial queries become hot topics in current research. In the version-increment based spatio-temporal data model, "Version" is used to represent states of geographical phenomena, while "increment" changes of spatio-temporal phenomena based on version. The geometrical space in the version-increment spatio-temporal data model can't be divided as it is the atom of space and time, and instead it can only change in forms of occurring, disappearing, replacing and so on. Occurring and disappearing are natures of changes, which can be integrated into basic types of spatio-temporal changes such as combination, division, replacement and so on. One feature may be changed in its geometrical space or attributes many times during its life-time. The geometrical space is considered as the primary atom of spatio-temporal index and it only has two changes during its life-time, with the first occurrence or replacing, the second disappearance or being replaced. The snapshot at any time is equal to the addition of the former snapshot and the increment between them. The amount of increment will grow larger and larger as time passes by, which makes it rather difficult to search for the increment between the version and some snapshot, thus there is a high demand for building spatiotemporal indexing mechanism based on increment. This paper proposes the version-increment based spatiotemporal data model by abstracting existing spatiotemporal data models and put forward the snapshot-increment spatiotemporal indexing mechanism for this model. This mechanism takes both space and time as equally important dimensions and makes it able to perform quick searches based on time points and intervals, so efficiency of spatial and temporal queries are both concerned about.

1. INTRODUCTION

Spatio-temporal index is one of the key technologies applied to spatio-temporal database engines. Current research of spatio-temporal indexes is just at its primary stage (Huang,2002), which mainly represents as extensions of traditional time indexes or spatial indexes. The former takes time as an independent dimension, with space just an attribute of time indexes, while the latter takes space as an independent dimension, with time just an attribute of spatial indexes.

Several practical methods have been put forward for multi-dimensional spatial indexes at the present, such as R tree, R+ tree, k-d tree, k-d-B tree, Grid structure and improved versions of these methods. However, efficiency of temporal queries is neglected. On the other hand, methods of indexes based on valid-time or transaction-time have been also proposed for queries of historical information, at the same time efficiency of spatial queries is neglected also. Therefore, methods for spatio-temporal indexes which consider efficiency of both temporal queries and spatial queries become hot topics (Luo,2002) in current research. There are four typical indexes (Theodoridis,1998; Liu,2003) which take both temporal and spatial factors into account, that is MR-trees、HR-trees、3D R-tree and RT-trees. MR-trees and HR-trees use R-tree structure to organize spatial information at any time and save spatial distributions (Liu, 2003) of objects at different time, which change to tree structures (Theodoridis, 1998) when the number of moving objects becomes larger. HR-trees creates a R-tree index (Luo,2002) for each time stamp, which shows good performance for time slice queries but bad for interval queries because the same data item could be retrieved many times. 3D R-trees takes time as an independent dimension with objects forming a cubic if they keep static in a period, so it causes great reduction in efficiency of indexing. RT-trees is a spatio-temporal version of R-tree, in which temporal and spatial information are separately maintained. This method has many limitations in that many index items will be created with RT trees expanding rapidly (Zhang, 2003) if there are large amounts of objects in change. The biggest weakness of R-tree is that MBR may overlap with each other which leads to repeated retrieval for the same path and thus affect efficiency (Luo, 2002) in retrieval.

In this paper, we put forward a mechanism for snapshot-increment based spatio-temporal indexes after exploration of the version-increment spatio-temporal data model, which pays attention to efficiency in both temporal queries and spatial queries by regarding time and space as dimensions of equal importance in spatio-temporal indexes. In the final section of the paper, the spatio-temporal queries, which the spatio-temporal indexing mechanism is applied to, are implemented based on version-increment spatio-temporal model.

2. THE VERSION-INCREMENT BASED SPATIO-TEMPORAL DATA MODEL

2.1 The Spatio-temporal Data Model Based On Version-increment

In the version-increment based spatio-temporal data model, “Version” is used to represent states of geographical phenomena, while “increment” changes of spatio-temporal phenomena based on version. This model can be regarded as the abstraction of Basic States Modification Model, Event-based Spatio-temporal Data Model (Peuquet, 1995), and Space-time Integration Model, and theoretical basis of these three models as well. The former two models suit raster data, while the latter one splits geometrical space into segments of arcs. The new model chooses spatial features as its object basing operations on vector data. “version” is a state of geographical phenomena at a certain time stored in databases; “snapshot” is a instant state of geographical phenomena obtained by operations such as querying, interpolation among versions. “version” is called in terms of database storage; “snapshot” is said in terms of operations in databases, these two terms are the same in nature. This paper employs the version-increment based spatio-temporal data model and the spatio-temporal indexing mechanism based on snapshot-increment, to differentiate concepts of spatio-temporal data models and spatio-temporal indexes.

The geometrical space in the version-increment spatio-temporal data model can’t be divided as it is the atom of space and time, and instead it can only change in forms of occurring, disappearing, replacing and so on (Fig.1). There are two atoms of objects in geometrical space, one an ellipse, the other a rectangle. At the T1 time, a triangle appears drawn in solid lines; at the T2 time, the ellipse dies drawn in broken lines; at the T3 time, the rectangle which dies and drawn in broken lines, is replaced for a pentagon which appears and drawn in solid lines.

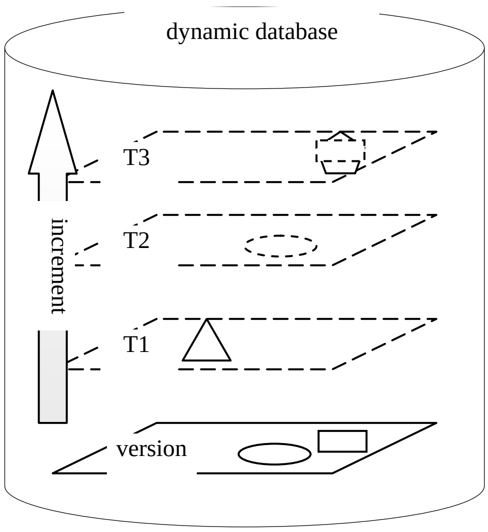


Figure 1.a

Occurring and disappearing are natures of changes, which can be integrated into basic types of spatio-temporal changes such as combination, division, replacement and so on. Division means a feature dying with many features in the same place appearing; combination means many features dying with a feature in the same place appearing; replacement means many features dying with many features in the same place appearing.

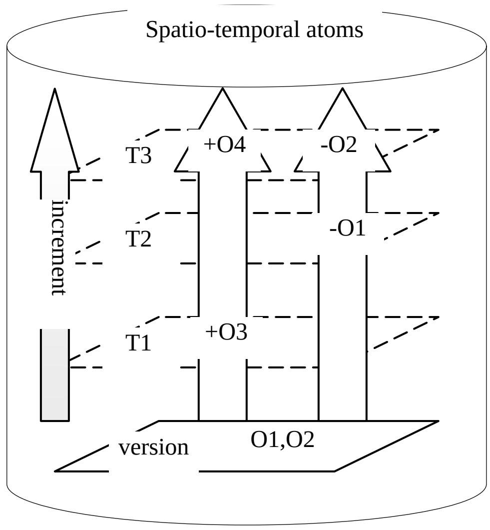


Figure 1.b the Spatio-temporal Data Model and Index Based on Version-Increment

2.2 The Storage Structure for Spatiotemporal Data

Changes of features include changes in both space geometry and attributes, so three tables, i.e. the table of space geometry, the table of features, and the table of attributes, will be created to save geographical information for the convenience of storing spatio-temporal data(Tab.1 1～3). The table of space geometry includes space geometry’s identifier, space geometry’s shape, space geometry’s life-time and features’ identifier. The table of attributes includes features’ identifier, attribute name, attributes’ life-time. The table of features includes features’ identifier, the first space geometry’s identifier, and features’ life-time.

Tab.1 Space Geometry of Features

space geometry’s identifier	space geometry’s shape	space geometry’s life-time	Features’ identifier
Geometry1			Feature1
Geometry2			Feature1
Geometry3			Feature2

Tab.2 Features

Features’ identifier	the first space geometry’s identifier	features’ life-time
Feature1	Geometry1	(T_{11},T_{12})
Feature2		(T_{21},T_{22})

Tab.3 Attribute of Features

Features’ identifier	Attribute 1	attributes’ life-time 1	……	Attribute n	attributes’ life-time n
Feature1	Attr1			Attrn	

When a feature has many geometrical spaces with time sequences, only the object which appears first in the geometrical space should be recorded in the table of features, while other objects will be obtained by analyzing both the "features'identifier" field and the "space geometry's life-time" field. The size of the table of features won't change when a feature's geometrical space changes a great many of times. The features' identifier can be used to search for all the objects in the feature's geographical spaces and sort them by the start-time of the life-time of these geometrical spaces. Objects in the geometrical space with the same start-time will appear in the same time; objects in the geometrical space with the same end-time will disappear; objects in the geometrical space with one's start-time the same as the other's end-time will in a change of replacement. This relating mechanism could also be applied to complex features, features with many objects in the geometrical space. Each feature's geometrical space is stored in disks sorted by the time sequence to get quick access to each feature's objects in the geometrical space.

The version-increment spatiotemporal model shares advantages of both the Sequence Snapshot Model and the Basic States Modification model as an extension of these two models. This nature could be shown as follows: (1) It absorbs advantages of the Sequence Snapshot Model which refer to its intuitional and simple sequences, and so it is just some original representation of changes of geographical phenomena in terms of time, which makes it quite easy to determine characteristics of geographical phenomena; （2）In this model increments could be related to each other by the "features' identifier" field and the "geometrical space's life-time" field with spatial objects' topological characteristics in the time dimension involved, and therefore it overcomes the shortage of no association between objects in the Sequence Snapshot Model; (3) It also employs advantages of the Basic States Modification Model by only storing changes of geographical phenomena, which reduces redundancy of data, makes it simpler and more efficient to perform time-dimension based queries and analyses, so rules of changes of spatial objects could be easily obtained; (4) It overcomes the shortage of the Basic States Modification Model by taking vector data as its data source.

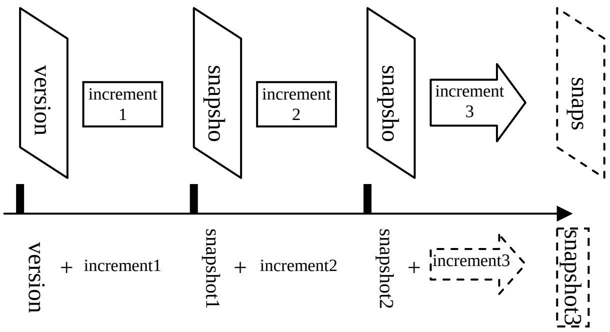


Figure 2. Relationship among version, increment and snapshot

3. SNAPSHOT-INCREMENT BASED SPATIO-TEMPORAL INDEXING MECHANISM

3.1 The Basic Principles

One feature may change its geometrical space or attributes many times during its life-time. The geometrical space is considered as the primary atom of spatio-temporal index and it only has two changes during its life-time, with the first occurrence or replacing, the second disappearance or being replaced. The geometrical space can be plus or minus which is to say that both occurrence and replacing refer to the plus sign (+) while both disappearance and being replaced refer to the minus sign (－). The geometrical space in the version should be assumed to be neither plus nor minus (Fig.1). Therefore increments of each object of the geometrical space will appear in the form of both a plus atom and a minus atom. The snapshot at any time is equal to the addition of the former snapshot and the increment between them, i.e.:

Current snapshot ＝ the former snapshot + the increment

Version is the original snapshot (Fig.2).

The snapshot can be seen as the result of logical operations performed on atoms of the geometrical space. In Fig.1 O1 and O2 respectively represent the ellipse and the rectangle in the version; +O3 refers to the appearance of the triangle in the increment; -O2 and +O4 show the replacement of the ellipse with the pentagon (Tab.4).

Tab.4 Logical sum between space geometry atoms

T0	Version	:	{O1, O2}	This version only have two geometrical objects: O1, O2
T1	Increment	:	+ O3	O3 appears
T2	Increment	:	- O1	O1 disappears
T3	Increment	:	- O2，+O4	O4 replaces O2
T3	snapshot	:	{O3，O4}	The result of these logical operations

The amount of increment will grow larger and larger as time passes by, which makes it rather difficult to search for the increment between the version and some snapshot, thus there is a high demand for building spatiotemporal indexing mechanism based on increment.

3.2 The Nodes of The Spatiotemporal Index

The spatiotemporal double-field node is used in the snapshot-increment spatio-temporal index. The node which embodies characteristics of space and time includes two fields: one is the set of geometrical spaces; the other the time interval (Fig.3). The set of geometrical spaces can be divided into three groups: the set of identifiers of the appearing geometrical space, the set of identifiers of the disappearing geometrical space, and the snapshot at the start-time of the time interval. The former geometrical space and the latter geometrical space which are involved in the change of replacement, correspond to the disappearing set and the appearing set respectively. Increments between geometrical spaces in a time interval can be represented by these two sets.

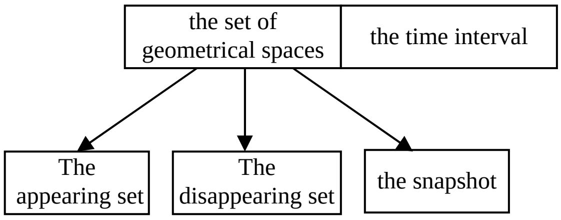


Fig.3 The node with spatiotemporal double fields

3.3 The Mechanism of the Spatiotemporal Index

Tree structures can be created based on nodes with double fields (Fig.4). The snapshot-increment spatiotemporal index can determine the depth of trees by the granularity of time and balance the load of nodes at the same level automatically. Here “balance” means controlling the size of the set of geometrical spaces by adjusting the length of the time interval. The number of nodes depends on the depth of trees and the length of the time interval of nodes.

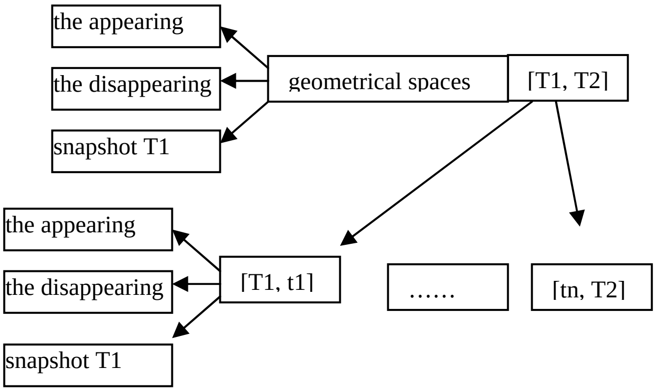


Figure 4. The spatiotemporal index structure based on snapshot-increment

4. THE EXPERIMENT

4.1.1 The Formal Description

Logically speaking, the snapshot is the set of valid identifiers of features at a set time, i.e.:

$$snap(t) = \{ featureID_{[i]} \mid t \in [startTime , endTime] \}$$

Here, snap(t) denotes the snapshot at time t, which can be empty. The increment, changes between snapshots, logically is the union set of appearing operations and disappearing operations at a set interval, i.e.:

$$\Delta[t_1,t_2] = \{+featureID_{[i]} \mid operTime \in [t_1, t_2]\} \cup \{-featureID_{[i]} \mid operTime \in [t_1, t_2]\}$$

where, $\Delta[t_1,t_2]$ denotes changes during the time interval $[t_1,t_2]$, and “operTime” does the time when the feature changes. Both of the snapshot and the increment are taken as the set of identifiers of features, and to some extent a snapshot is an increment with the plus sign. The snapshot-increment index can be thought of as logical operations between sets of features and sets of operations, which take sets of features of snapshots as results, i.e.:

$$snap(t_2) = snap(t_1) + \Delta[t_1,t_2]$$

4.1.2 The Preparation For Data

This experiment studies the land subdivision (Fig.5), organizes data using the version-increment spatio-temporal data model and implements queries for snapshots basing on the snapshot-increment spatio-temporal indexing mechanism.

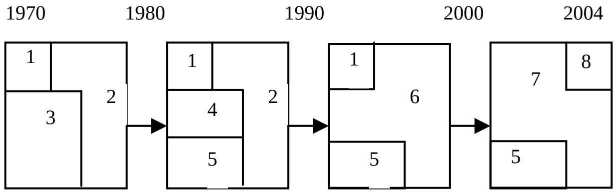
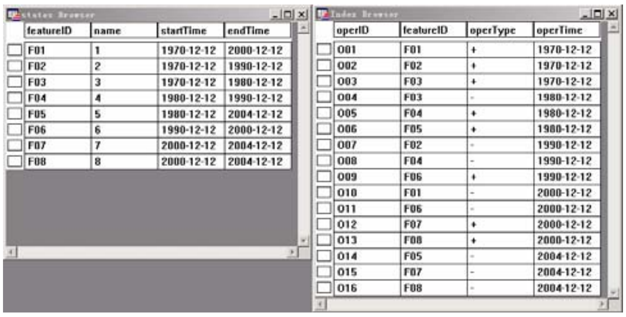


Figure 5. Process of land subdivision

The MapInfo table is used to manage states of parcels, whose structure includes the parcel’s identifier, the name of the parcel, valid start-time and end-time of the parcel. And meanwhile the MapBasic language is used for creating index tables based on the table of states (Fig.6). For simplification, in a table of states one identifier of features corresponds to one identifier of geometrical spaces. And the index table is composed of four fields such as operID, featureID, operType and operTime in the following figure. Index tables are sorted by the time when operations occur, whose operType can have the plus sign or the minus sign, respectively meaning appearance or disappearance.



featureID	name	startTime	endTime
F01	1	1970-12-12	2000-12-12
F02	2	1970-12-12	1990-12-12
F03	3	1970-12-12	1980-12-12
F04	4	1980-12-12	1990-12-12
F05	5	1980-12-12	2004-12-12
F06	6	1990-12-12	2000-12-12
F07	7	2000-12-12	2004-12-12
F08	8	2000-12-12	2004-12-12

operID	featureID	operType	operTime
001	F01	+	1970-12-12
002	F02	+	1970-12-12
003	F03	+	1970-12-12
004	F03	-	1980-12-12
005	F04	+	1980-12-12
006	F05	+	1980-12-12
007	F02	-	1990-12-12
008	F04	-	1990-12-12
009	F06	+	1990-12-12
010	F01	-	2000-12-12
011	F06	-	2000-12-12
012	F07	+	2000-12-12
013	F08	+	2000-12-12
014	F05	-	2004-12-12
015	F07	-	2004-12-12
016	F08	-	2004-12-12

Figure 6. States and Index about spatiotemporal data

The method for creating index tables: each feature in the table of states has two records sorted by the time when operations are performed in the indexing table. One records the start-time and represents the appearance of the feature with corresponding operation type “＋”, while the other records the end-time and represents the disappearance of the feature with corresponding operation type “－”. Here, the end-time of active features can be replaced by the current time.

4.1.3 The Spatiotemporal Queries

The spatio-temporal index is built for the efficiency of spatiotemporal queries and the convenience of spatio-temporal analyses. Spatiotemporal querying is divided into snapshot querying and evolvement querying. Traditional spatiotemporal models focus on semantic descriptions of space and time, but neglect efficiency of spatiotemporal queries. In the light of the formal description of the index and characteristics of the indexing table, the snapshot query is made up of two parts: one the last snapshot, the other the increment. When the snapshot searched for is the original one, the last snapshot will be empty. The increment query is performed according to the time of current snapshot t_2(1979-10-10) and the time of the last snapshot t_1; if the last snapshot doesn’t exist, time t_1 is the initial valid time of the database, here t_1 is 1970-12-12 (Fig.7). In the indexing table records during time t_1 and t_2 are organized into identifiers of features with the plus or minus sign, and these identifiers and identifiers of features of snapshot t_1 are involved in the logical sum operation which affects efficiency of querying. This experiment performs the "logical sum" operation by operations of the snapshot tables. Firstly, the state of the very snapshot just before the snapshot searched for is used to initialize the snapshot table. Then identifiers of features with signs are extracted in the indexing table, and some operations will be performed following the below rules: Records of the identifier of features will be added when the sign is plus, and records will be deleted until all the records in the indexing table are searched for when the sign is minus. Finally, the result of the snapshot table is the feature of the snapshot searched for.

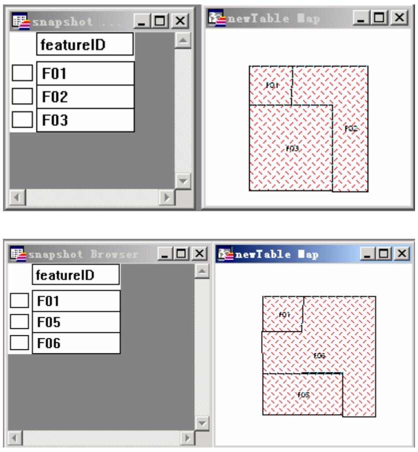


Figure 7. Original snapshot and successive queried snapshot

Feature's evolvement querying refers to the search for changes happening in single feature F during the set period $[t_1,t_2]$. Parcels in change have these two characteristics as follows: (1) parcels before changes and after changes have adjacent relationship in time, i.e. the end-time of parcels before changes is just the start-time of parcels after changes. Correspondingly records of parcels before changes and after changes in the indexing table are also adjacent, which can help find the set of increment △F. For example, F3 at the time 1980-12-12 is divided into F4 and F5; operations such as "-F3", "+F4", and "+F5" share the same time , and also are adjacent in the indexing table. (2) As for parcels before and after changes, their spatial positions stay the same, but they have overlapping relationship in the space, by which sets of features in change in feature F can be obtained from △F. As △F decreases the amount of features involved in the overlapping spatial relationship, a great deal of time is dramatically saved.

5. CONCLUSION

This paper proposes the version-increment based spatiotemporal data model by abstracting existing spatiotemporal data models and put forward the snapshot-increment spatiotemporal indexing mechanism for this model. This mechanism takes both space and time as equally important dimensions and makes it able to perform quick searches based on time points and intervals, so efficiency of spatial and temporal queries are both concerned about. In addition to the above advantages, there are many other merits: (1) Changes are taken as the object in index mechanism, which is based on operations of theses changes; (2) It turns the spatio-temporal index to temporal linear index, which can build multi-level indexes with the help of spatial analysis; (3) The indexing table can be created automatically; (4) The process of indexing is the process of searching and logical operations; (5) Double-direction queries can be performed. Signs of operations in opposite-direction queries will be on the opposite; (6) Data are separated from indexes. This experiment just studies the land subdivision, therefore there is necessary to extend this snapshot-increment spatio-temporal index to other areas for application.

References:

Huang, S., Peng, J., 2002. Research of Index Mechanisms in Discrete Spatio-temporal Environment. *Journal of Soochow University(Engineering Science Edition)*, 22(5):7-12.(in Chinese)

Luo, X., 2002. Spatio-temporal Object Model and Application in Cadastral Information Systems. *Wuhan:Wuhan University Publication*.

Theodoridis, Y., Sellis T., 1998. Papadopoulos A,et al. Specifications for Efficient Indexing in Spatiotemporal Database. *Capri:Proceedings of SSDBM'98(IEEE)*.123-132

Liu, J., Yang, X., et al., 2003. A Spacialtemporal Index Based on R*-tree.*Computer Engineering*, 29(14):60-62. (in Chinese)

Zhang, S., Yang, Z., 2003. A Spatio-temporal Indexing Method for GIS. *Geomatics and Information Science of Wuhan University*, 28(1):51-54.(in Chinese)

Peuquet, D., Duan, N., 1995 An Event- based Spatiotemporal Data Model(ESTDM) for Temporal Analysis of Geographical Data. *International Journal of Geographical Information System*, 9(1): 7-24.