A DATA STRUCTURE FOR SPATIO-TEMPORAL INFORMATION MANAGEMENT

Y. Ohsawa *, W. Guo, M. Sakurai [a]

Department of Infomation and Computer Sciences, Faculty of Engineering
Saitama University, Shimo-okubo, 255, Saitama, 338-8570, Japan
ohsawa@mm.ics.saitama-u.ac.jp

KEY WORDS: Spatio-temporal, Implicit topology description, STIMS, Free GIS, GBD tree

ABSTRACT:

This paper describes about spatio-temporal geographic information system, named STIMS(Spatio-Temporal Information Management System). STIMS handle temporal information assigned each geographic object and can execute several kinds of spatial operations on the world of specified time. The system adopts implicit topology data model to handle such temporal data. By this data model, the data structure for spatio-temporal geographic objects becomes very simple. This paper describes about the data structure and how to handle spatio-temporal data on the system. STIMS is now released as free software on the WEB.

1. INTRODUCTION

On geographic information system, to handle transition of real world, time slice or version management of whole map have been adopted. On this method, the change of each geographic entity cannot be described. The change is known to compare two temporally neighbouring time slice maps. On the other hand, if time stamps which shows the generation and disappearance time are assigned for each entity, we can catch the world of arbitrary specified time.

If isolated geographic object, for example building or landmark are treated, assigning the time stamps to each entity is achieved very easily. On the other hand, when division of a parcel is concerned, this time management is not trivial. Because the topology of the neighbour of the parcel has been changed by this division. On conventional geographic information systems, region have been managed as network composed by nodes and links. Then, if a parcel is divided into two, this affects to the organization of the network.

On usual geographic information systems, topological information has been described explicitly using relational tables or pointers[1]. This method was very natural and optimum choice when GIS came into existence (1970s). Because the computer processing speed at that time was very slow and the amount of main memory was limited. Then, to describe topological information explicitly was mandatory for obtaining good response.

However, during later two or three decades, computing speed has increased more than 1,000 times. By this computing power progress, new methodology which restore the topological information by calculation when it is necessary becomes actual[2],[3]. We call this methodology as implicit topology description method.

Not only for region data, but also for network data, like road and river, topology description is very important. When we want to know the shortest path between two specified points (start and destination) on a road network, the path can be found by successive following the connected road segments under the control of A* algorithm[4] or Dijkstra algorithm[5]. When the topological information of road network is given explicitly, the set of connected road segments at an intersection can be known very easily by following pointers or inquiring relational tables. On the other hand, on implicit topology model, the topological information is restored by spatial retrieval at the each intersection. The connecting road segments at an intersection have the same end point, then these road segments can be found by a spatial search to find the road segments whose one of the end point meets the intersection location. This type of spatial retrieval can be done efficiently using spatial data structure, for example R-tree[6] or its descendants[7]. In STIMS, GBD-tree[8],[9], which was developed the same purpose, is used for this spatial search.

By possessing topological information implicitly, temporal information management becomes easy[10]. For example, suppose the situation to manage the transition of administrative borderline for 50 years time span. During the time span, cities may have been merged and split, and the names of the cities may have been changed. When explicit topological expression is used for managing these changes, the structure becomes very complicated like spaghetti. On the other hand, on implicit topology description model, the data structure stays very simple despite such changes. Because, city border at specified time is restored by spatial retrieval using spatio-temporal key.

A defect of implicit topology model is in the longer calculation time, because the topology is always restored by calculation. To solve this problem, STIMS adopts topology cache method[11]. Once topology is restored, the restored topology is kept in cache for the next use. By this method, calculation time reduced drastically. When a part of data is changed, the cache data which is affected by the change is discarded, then the topology is re-calculated. When the time view is changed, the cache which affect the time view change is also discarded.

STIMS is implemented by C++ language on Windows PC and it is released as free software.

In section 2, data model used in STIMS is described. In section

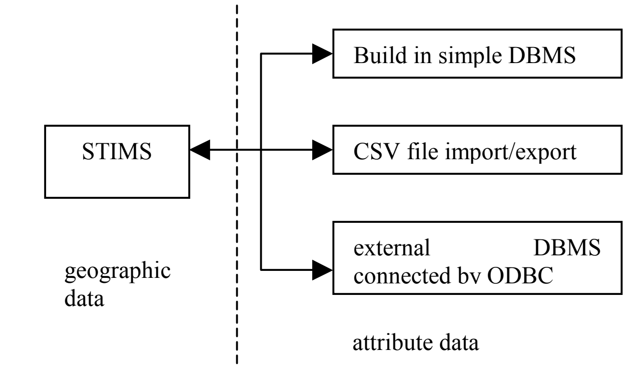


Fig.1 STIMS and DBMS for attribute data

3, several spatial operations to restore topological information is described. In section 4, temporal data handling method and its data structure is described. GIS refers several dataset, including spatial entity file, attribute data file, several definitions and legends. Such kinds of data formats are described in section 5. Section 6 concludes the paper.

2. ORGANIZATION OF STIMS

Entities which managed in GIS are composed by geometric data and their attribute data. Mainly, the shape and the position of geographic entities are managed in STIMS. To handle attribute data, STIMS furnishes three methods as shown in Fig.1. One is built-in very simple DBMS, and the second one is CSV interface. The last one is ODBC interface; this one is the most useful and common interface.

To combine between geometric object and its attribute, unique `key' is assigned for both data. Usually, a unique numerical identifier (ID) is assigned for corresponding object on both systems. STIMS also assigns such ID for geometrically isolated entities and line entities. In conventional GIS, to manage regional data, TIGER or DIME structure is adopted. On these methods, the information denoting left and right region IDs is assigned for each borderline. While region is stable, this method works very well. However, when occasionally occurring changes have to be concerned, this method makes data management very complicated. Each line must have several IDs describing the history of changes.

To simplify the data management, STIMS separates attribute management from region shape. A point data located inside of a region is assigned for a region as a spatio-temporal key. The point has the location (X and Y axis position) and time data denoting the life span of the object. The shape of a region, in other word the topology of the borderline of the region, is restored when a region is specified for visualization or spatial analysis. While the restoration process, time information assigned for each line segment is referred. Using only valid line segments at the specified time, the region shape is restored.

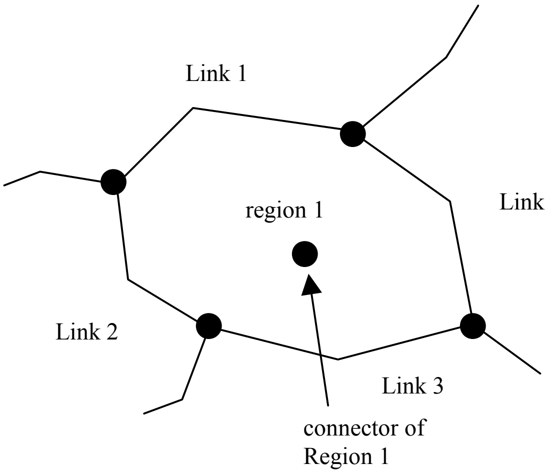


Fig 2. Region definition

Figure 3 shows a part of a network, such as road network. In this figure, three roads, L1, L2 and L3 are connected each other at node N. In the network, roads correspond to links and the intersections correspond to nodes. The explicit topology model, which has been used usual GIS, deals with the topology of the network explicitly using pointers or relation tables.

The implicit topology model, on the other hand, describe topological information by definition and method functions. This method restores the topological relation through spatial operations. Namely, other links connected to the specified link are found by a spatial retrieval specifying the end point of the link. The spatial retrieval returns the links of which one endpoint meets at the same position with the specified end point. In this paper, this operation is called as 'node expansion'.

To perform such an operation at high speed, a special data structure, so called spatial data structure, is necessary. Several kinds of spatial structures, including R-tree[6], R*-tree[7] and GBD-tree[8], have been proposed. In this paper, GBD-tree, which is developed one of the authors, is used to manage spatial data. The algorithms described in this paper can be widely applicable to other types of spatial data structures.

In the implicit topology model, the attributes and the geometrical shape of the entities are managed separately. When correspondence among them is necessary, they are combined by spatial retrieval. Figure 2 shows a line object (e.g. road), and the object has attributes (e.g. name of the road, width of the road, number of traffic lanes). Usually, when a GIS displays the entities on CRT, all attributes data are not always necessary. Thus, it is sufficient that the attributes can be retrieved when they are requested. Then, if the shape and the attributes are separated, the time required to read the entities from a hard disk can be shortened. For the reason mentioned above, it is preferable that the shape and the attributes are managed separately even on explicit topology based GIS.

The shape and the attributes are related to each other by their position. Lines and regions have extent in two-dimensional space. Then, to determine their positions, a unique point, or representative point of the object is necessary. We determine the point as the center of the minimum-bounding rectangle (MBR) of the object. The attributes are located at the

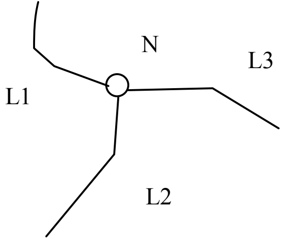


Fig.3 Network node and links

representative point. Hereafter, we call the representative point the 'connecter', in the sense that it connects the shape and the attributes.

To generate the relationship between the shape and the attributes, two inverse directional operations are necessary: (1) find the attributes (connector) from the shape, and (2) find the shape from the connector. The details of these operations are described in section 3.

Regions are expressed by a set of borderlines, as shown in Figure 2. Each closed area divided by the borderlines is a region. In the implicit topology model, all borderlines are also not mutually explicitly related. Hence, the shape of a region must be also restored by using node expansion.

The attributes of a region are related to the geometrical shape by the connector. In the case of region, the connector is placed somewhere inside the region. Because the shape of region is varied, it is not easy to determine a rule for placing the connector. For this reason, the position of the connector is restricted only to somewhere inside the region, as shown in Figure 3. To determine the attributes of a region, the following steps of operations are necessary: (1) restore the shape of the region, (2) find the position of the connector by searching inside the region, then (3) read the attributes related to the connector.

Summarizing the operations used in the implicit topology GIS, the followings basic generations are mainly necessary:

(1) node expansion
(2) generation of correspondence between the shape and the connector of line objects
(3) generation of correspondence between the shape and the connector of region objects

By combining these basic operations, more complicated geographical operations, such as the shortest path finding, are realized.

3. OPERATIONS TO RESTORE TOPOLOGY

3.1 Spatial operation for restoring topology

Figure 3 showed a simple example of topology. In explicit topology description, relation tables or pointers describe the connective relationships among L1, L2, L3 and N, allowing us to easily recognize the links that are connected to each other by referring to them. On the other hand, a GIS based on implicit topology description must restore the relation by spatial retrieval.

In STIMS, GBD-tree is used for this purpose. The GBD-tree is a multi-dimensional data indexing structure. Figure 4 shows the structure. As shown in this figure, each node has M slots. Each slot in non-leaf nodes has a pointer to the child node, its minimum-bounding rectangle (MBR) and the center point of the MBR for indexing. The MBR is the tightest rectangle bounding all entities contained in offspring nodes. A slot in the leaf nodes has pointer to the entity, MBR of the entity, and the layer of the entity. Spatial retrieval can be done efficiently by referring to the MBR.

As mentioned in Section 2, node expansion is a basic operation for restoring topology. This operation can be performed by searching the links of which an endpoint meets the position of the node. Following shows the procedure of searching the links connected to node N on GBD-tree.

(Step.1) Let T be the root node.
(Step.2) If T is not a leaf node, each offspring node E is checked whether the MBR of E includes the location of node N. For all including nodes, search the links connected to node N on the sub tree whose root node is E.

(Step.3) If T is a leaf node, all entities included the node are checked whether an end point of it meets the position of node N. If so, the entity is connected with node N.

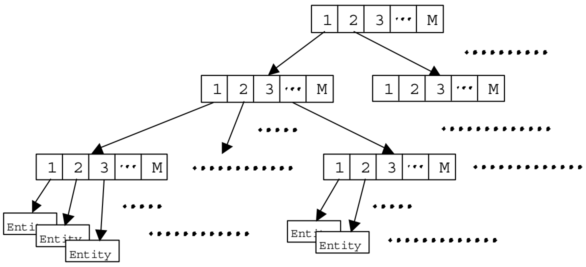


Fig.4 Hierarchical structure of GBD-

3.2 Searching connector of a specified line

The connector of a line object is found by searching the connector object corresponding with the center of the MBR of line object on GBD-tree, because the connector is located at the center of the MBR. In the following sentence, we describe the procedure to find the connector of link L on GBD-tree, when link L is specified.

(Step.1) Let T be the root node.

(Step.2) If T is not leaf node, each offspring node E is checked whether the MBR of E includes the center of the MBR of L. If so, for all including nodes, search the connector of L on sub-tree whose root node is E.

(Step.3) If T is a leaf node, check all connectors under the node whether its location accords to the center of the MBR of L.

3.3 Searching connector of a specified region

To find the connector of the region that includes the specified point, it is necessary to restore the region, and to find the connector located in the region.

In the implicit topology model, a region consists of a boundary and a connector (the representative point). The boundary consists of a collection of line segments (links). The connector is placed inside the region, and is used to connect the attribute information of the region.

Region restoration is a procedure to restore the shape of a region. This procedure is necessary in several operations including to restore the shape to display and to know the attribute of the region. This procedure is written as follows.

(Step 1) Find one of the boundary line segments of the specified region. This can be done by retrieving the rightmost boundary line segment on the horizontal line stretched from the specified point P, as shown in Figure 5. Let the found line segment S. This operation can be executed efficiently on GBD-tree.

(Step 2) Follow the line segments forming the specified region counter-clockwise from S to return back to S. At the end of S, the next segment can be found as follows. In Figure 6, let link L be a boundary line segment of which either end is N. Expand the node to find all line segment meeting at N. Then, determine the line segment located counter-clockwise neighboring to line L. Repeat the same operation at the end of the opposite side end of the segment until return back to link L. After the operation, all line segments consisting of the region are found.

Figure 5 summarizes the operations.

Then to find the connector of the region R that includes the specified point, it is necessary to find the connector located in the region R on GBD-tree. This procedure is written as follows.

(Step.1) Let T be the root node.

(Step.2) If T is not leaf node, each offspring node E is checked whether its MBR overlapped with the MBR of R. If so, for all overlapping nodes, search the connector of R on sub-tree whose root node is E.

(Step.3) If T is a leaf node, each connector under T is checked whether it is located in R.

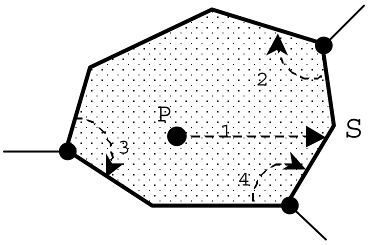


Fig.5 Region restoration

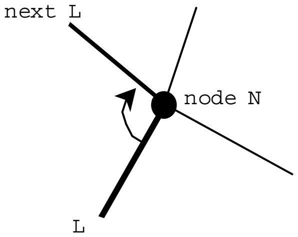


Fig.6 Node expansion

3.4 Searching line entity corresponding to a specified connector

When some attributes are specified to display the line segments meeting the conditions, the connector that satisfies the condition are retrieved. The shape of the line is searched by spatial retrieval in the way the center of the line locally corresponds to the connector.

This retrieval can be performed on GBD-tree as follows.

(Step.1) Let T be the root node.

(Step.2) If T is not leaf node, each offspring node E is checked whether its MBR include the location of the connector. If so, for all including nodes, search the line on sub tree whose root node is E.

(Step.3) If T is a leaf, each entity under the node is checked whether the center of its MBR according to the location of the specified connector. If so, it is the line object according to the specified connector.

3.5. Searching region corresponding to a specified connector

The connector of a region is located in the region, so when a connector is specified to restore the region, the procedure can be done by the same procedure described in 3.3.

4. SPATIOTEMPORAL RETRIEVALS

In GIS which deals with discrete temporal events, the types of retrievals are categorized to the following:

(1) Spatial retrieval over the present data

(2) Spatial retrieval over at a specified time data

(3) Detection of the difference between two specified times.

The first type of retrieval is not temporal retrieval. Conventional GISes usually execute this type retrieval. The proposed data structure can also execute the retrieval, just not using the extension for temporal data, as described in [10]. There is no retrieval time loss by the extension. In the following two subsections, the other two types of retrievals are described.

4.1 Expansion of GBD-tree for temporal data

Basic idea how to manage spatio-temporal data is described in [10]. The method uses geographic differential script file (GDSF) to record past data. Then, this paper describes about abstract feature of the data structure. The largest difference between the original GBD-tree and the structure for spatio-temporal data is that the latter has a priority queue attached to every leaf node to store past data.

GBD-tree uses different mechanisms when to register data and to retrieve data. The characteristic is very important when to insert temporal data. The leaf node in which the temporal data is to be inserted is determined depending on the center position of the data. On the other hand, the node is not determined uniquely on R-tree and its successors because the data are divided by only the shape of the MBR at the time that the division is invoked.

The priority of the queue is ordered by the time-print from new to old. To distinguish the MBR of temporal data from that of present data, let the MBR of the present data be denoted as MBRp, and the MBR of the temporal data be denoted as MBRt.

Retrieval from the queue is also required in order to restore the old timed data. In order to avoid omission of temporal data during spatial retrieval, each node of the GBD tree has the MBR of temporal data. When restoring the map of a specific date, spatial retrieval is executed, referencing the MBR of temporal data attached to each node. If the MBR of temporal data overlaps the specified retrieval area and if the node is a leaf node, then the queue storing temporal data is reviewed and applied to the command that is controlling the current data set.

4.2 Spatial retrieval of specified time data

The most typical spatial retrieval is range retrieval. Other spatial retrieval (for example, to find nearest neighbor) can also be executed by a combination of range retrieval. The following describes how to do range retrieval on the proposed data structure.

First, retrieval of the present data is done by the usual method of searching the GBD tree. Specifically, the overlap between the specified retrieval area and the MBRp on each node is inspected. If these two are overlapping, descend the tree and repeat the same check for all the child nodes. If the node is a leaf node, select the entities actually included in the specified range and add the entities to the result set. In this searching process, there is no overhead time caused by adding temporal information to the tree structure.

When a particular time (old date) is specified outside of the retrieval range, the priority queues attached to the leaf nodes are inspected. Range retrieval of a specified time is executed by the following steps, let the specified time be T and the specified range be R.
(1) empty the result set S.
(2) check whether MBRp or MBRt overlaps with R
(3) if they are overlapping and the node is not a leaf node, descend the tree and repeat the check for all child nodes.
(4) if they are overlapping and the node is a leaf node, copy the data in the leaf node to a working space W.
(5) apply the GDSF commands attached to the node whose time print is not older than T.
This operation is executed on the data in the working space W. Add the result of W to S.

Fig.7 summarizes the steps of temporal retrieval. Every leaf node consists of two parts, the present data and the historical data queue. After having reached a leaf node, the present data part is copied to a working space W. Then the GDSF commands are applied until specified time. The contents in W are modified to the states of the specified time. Finally, spatial retrieval is performed on W.

The most important points of the method are in step (5). The first point is that the range retrieval with specified time is executed on a working space W, and there are no effects on the spatial data stored in the GBD tree. The second is that the influence of the operations by the GDSF commands is restricted within each leaf node. For example, a command to insert an entity has no effect on the other leaf nodes. When there is an insertion command on a priority queue of a leaf node, execution of this command has no effect on the results of the operation on the other nodes. In addition, if there is a deletion command on a priority queue of a leaf node, the object must exist on the same leaf node, because the GBD tree distributes entities to individual leaf nodes according to the center points. Also, deletion commands have no effect on other leaf nodes.

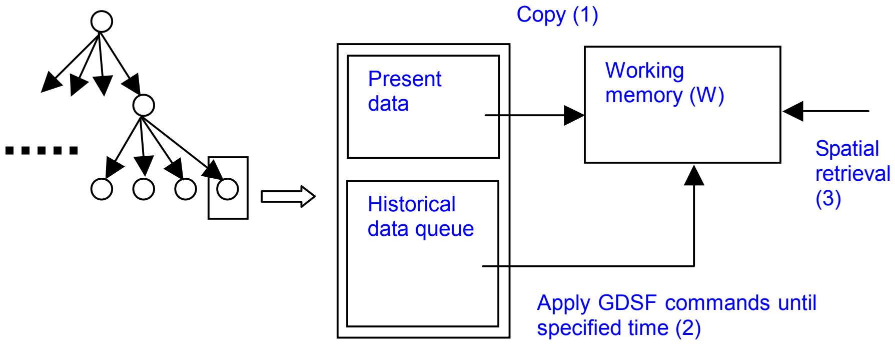


Fig.7 Steps of spatio-temporal retrieval

4.3 Detection of the difference between two specified times

The comparison of two specified points in time is a frequently requested operation in spatiotemporal GIS. The proposed data structure can execute this type of operation easily.

Let two specified points in time be t1 and t2 (t1 < t2: t1 is older than t2). This type of retrieval, the step (1) in Fig.7 is not necessary, because the state of the present has no effect on the result. Fist, skip the historical data queue until the time print is newer than t2. Then, execute the GDSF commands stored in the queue in the working memory W while the time print of the command is newer than t1. When the steps have completed, the entities existing in W are a part of the result. To do this operation over the leaf nodes overlapping the specified area, the change can be detected in the area.

5. DATA FORMAT

STIMS consists of several data files. In this section, the format of these files are described.

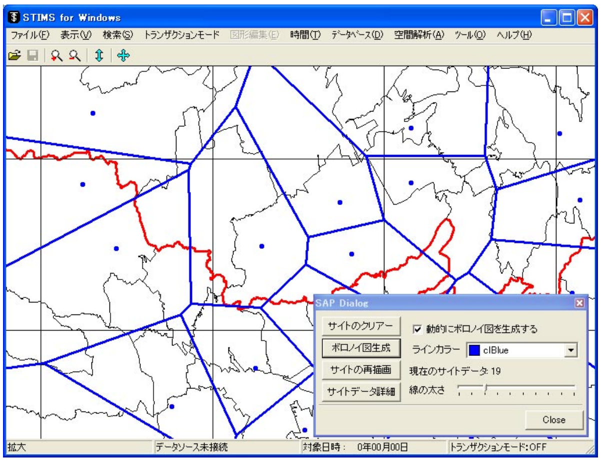
The *system definition file* is the most important file which describes all files used in STIMS and defines important parameters.

Spatial index file is for high-speed spatial retrieval, which is organized by GBD-tree. The contents of *vector file* and *connector file* are managed by this index.

Legend definition file defines several attributes of depicted line and regions, including the kind of hatching, kind of line, thickness and color.

Theme definition file define several kinds of theme used in several kinds of spatial retrieval. A theme is defined by a ordered set of line kinds. For example, city shape is defined the region divided by shoreline, country border, prefectural border, city border, and sheet border. The theme definition file defines such kinds of line for each theme.

Differential script file is for synchronization with external GIS. This file is similar with transaction file, recording addition and deletion of geographic entities.



☑ 動的にボロノイ図を生成する

ラインカラー: clBlue

現在のサイトデータ: 19

線の太さ

データソース未接続

対象日時： 0年00月00日

トランザクションモード:OFF

Fig.8 Voronoi diagram on STIMS

Current version of STIMS has simple DBMS to manage attribute information. The *attribute database file* showing in the figure is this file. For more complicated query, ordinary database managenet system, for example SQL server, MS ACCESS, or Postgre SQL, are used. To connect these attribute databases, STIMS provides two interfaces. The first one is *attribute database interface file*, and the other one is communication using ODBC. Suppose an order to color each city area to distinguish the population density according to a legend, the send to external attribute DBMS. Then the DBMS replay the result by an attribute database interface file.

6. STIMS IMPLEMENTATION

STIMS is now implemented on PC by VC++. At present, basic spatio-temporal data and attribute data handling is available. As spatial analysis functions, shortest path finding, Voronoi diagram are now implemented. On the STIMS map data, several kinds of multi-media objects, including image, movie, Word document, and URL for WWW can be embedded. Figure 8 shows an example of STIMS window. In this example, Voronoi diagram is displayed.

We are developing two kinds of implementation; one is stand-alone version, and the other one is for network environment. The latter one is implemented for JaMaPS viewer. JaMaPS[12] is an internet GIS viewer developed by KDDI Laboratory, Japan.

The implicit topology description model using in STIMS require much calculation time than conventional GIS. To solve this problem, we have developed topology cache method, and have implemented in STIMS already. By this topology cache, STIMS can work very fast.

REFERENCES

[1] Tor Bernhardsen, “Geographic Information Systems – An Introduction (second edition)”, John Wiley & Sons, 1999

[2] Shigeru Kakumoto, Michinori Hatayama, Hiroyuki Kameda, Tokihiko Taniguchi, “Development of Disaster Management Spatial Information System (DiMSIS)”, Proc. Of IEAS’97, 1997

[3] Shingo Otomo, Yutaka Ohsawa, “A Study on a No-topological Geographic Information System”, Proc. 8-th Functional Graphics Information System Symposium, pp.27-32, 1997 (in Japanese)

[4] N. J. Nilsson, “Problem-solving Methods in Artificial Intelligence”, McGraw-Hill, 1971 [3] Yutaka Ohsawa, Masao Sakauchi, “A New Type Data Structure with Homogeneous Nodes Suitable for a Very Large Spatial Database”, Proc. of 6th International Conference on Data Engineering, pp.296-303, 1990

[5] Alfred V. Aho, Jeffrey D. Ullman, “Foundations of COmputer Science”, p.483, Computer Science Press, 1992

[6] A.Guttman: R-trees: A Dynamic Index Structure for Spatial Searching”, Proc. ACM SIGMOD Int. Conf. On Management of Data, pp.47-57, 1984

[7] N.Beckmann, H.P.Kriegel, R.Schneider, B.Seeger, “The R*-tree: An Efficent and Robust Access Method for Points and Rectangles”, Proc. of the 1990 SIGMOD, pp.322-331, 1990

[8] Yutaka Ohsawa, Masao Sakauchi, “A New Type Data Structure with Homogeneous Nodes Suitable for a Very Large Spatial Database”, Proc. of 6th International Conference on Data Engineering, pp.296-303, 1990

[9] Yutaka Ohsawa, Masao Sakauchi, “A Proposal of Multi-Dimensional Data Structure with Two Kinds of Auxiliary Data for Retrievals and Managements”, IEICE Trans., J74-D-1, 8, pp.467-475, 1991 (in Japanese)

[10] Yutaka Ohsawa, Kim Kyungwol, “A Spatio-temporal Data Management Structure Based on Different Script”, Theory and Applications of GIS, Vol.8, No.1, 2000 (in Japanese)

[11] Hideki Nonaka, Yutaka Ohsawa, “An Implicit Topology Description for Network and Spatio-temporal GIS”, Proc. ISPRS2000, (CD-ROM Proceedings), 2000

[12] JaMaPS: http://www.jamaps.org

[13] Hideki Nonaka, Yutaka Ohsawa,”Processing Time Analysis on a GIS Based on Implicit Topology Model”, Theory and Application of GIS, Vol.9, No.2, pp.17-24, 2001 (in Japanese)