MULTI-LEVEL TRANSPORTATION NETWORK ADAPTABLE TO TRAFFIC NAVIGATION

Jun FENG
Graduate School of Engineering,
Nagoya University
Furo-cho, Chikusa-ku,
464-8603 Nagoya, Aichi, Japan
feng@watanabe.nuie.nagoya-u.ac.jp

Naoto MUKAI
Graduate School of Information Science,
Nagoya University
Furo-cho, Chikusa-ku,
464-8603 Nagoya, Aichi, Japan
naoto@watanabe.nuie.nagoya-u.ac.jp

Toyohide WATANABE
Graduate School of Information Science,
Nagoya University
Furo-cho, Chikusa-ku,
464-8603 Nagoya, Aichi, Japan
watanabe@is.nagoya-u.ac.jp

ABSTRACT

In a country wide Intelligent Transportation System (ITS), there is a need for processing road map and transportation information in many levels of details. In this paper, an integrated method for representing transportation information with multi-level road map is proposed. This method adopts a spatial index structure for managing map objects in multiple levels and uses an integrated method for representing the travel junctions (or traffic constraints), travel cost on road segments and turn corners. Based on the datasets generated by this method, queries in ITS applications can be responded efficiently in different levels of details.

KEY WORDS

Application development, transportation network, road map, spatial index, traffic constraint.

1 INTRODUCTION

The geographic database and the management facility/environment is one of the important subjects concerned with Intelligent Transportation System (ITS). A geographic database may contain static map data (including data of road network and other map objects), public transportation routes, and current travel cost (e.g., travel time) on segments of transportation network [1]. In a country-wide system, there is a need for processing the map and transportation information in many levels of details. This is due primarily to the desire of user to use (/see) relevant information only; too many details may hinder rather than help [2]. For example, a query of path search between two city halls belonging to two separated prefectures is solved more efficiently by dividing into three sub-queries: one sub-query for finding path between two cities based on the country-level and prefecture-level roads (i.e., highways and crossing-prefecture roads), and two sub-queries based on city-level roads (all the streets inside the city) for finding the path between the city halls and the upper level roads. Since the generalization of map information cannot be realized automatically [3], various methods have been proposed for maintenance of scaleless maps or multi-scales of maps [2, 3, 4]. However, transportation network is different from road maps. It is important to identify one-way roads with attributes of links, traffic constraints, information about turns between links, or access conditions from one link to another[5]. Moreover, for some important route planning problems, the turn cost, which is encountered when we make a turn on a cross-point, are also taken into consideration [6]. The traditional representation method represents transportation network with nodes and arcs. The traffic constraints and turn costs are managed by adding new nodes and arcs. The total number of nodes and arcs in the dataset is multiplied [6, 7]. Furthermore, as the neighbouring nodes in different level of details do not keep the same, the transportation information sets should be generated for every level separately. And there are two problems for the generated datasets: one is the complex maintenance process; and another is that it is difficult to support a path search based on the transportation networks in different level of details.

To solve these problems, in this paper, we propose an integrated representation method for the multi-level transportation network. Our method adopts a spatial index structure for managing map objects in multiple levels and uses an integrated method for representing the travel junctions (or traffic constraints), travel cost on road segments and turn corners. Based on the datasets created by this method, queries in ITS applications can be processed efficiently by using proper search methods.

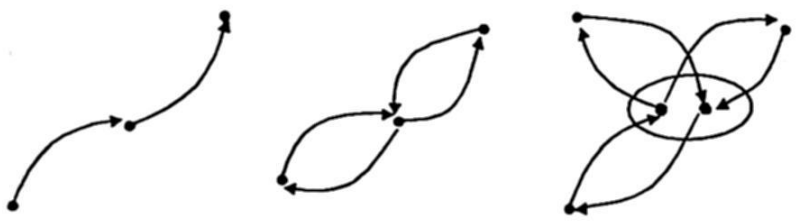
This paper is organized as follows. The related works for representation of multi-level road network and transportation network are presented in Section 2. The representation method for integrated management of multi-levels traffic conditions and spatial information about road network is proposed in Section 3 and Section 4 Section 5 analyses our method and Section 6 makes a conclusion on our work.

2 RELATED AND PREVIOUS WORK

The issue of this paper refers to the management of multi-levels of map and the integrated representation of transportation information and spatial information about road network.

2.1 INTEGRATED REPRESENTATION OF TRAFFIC INFORMATION AND ROAD NETWORK

To represent the traffic information on road network, a typical method [7] represents the road network using a directed graph. In the graph, each arc depicts a one-way road and each node corresponds to a junction. Two-ways roads can be presented as a pair of arcs: one in each direction. However, extra nodes should be added to the graph when there are any access limitations (constraints of specific traffic controls). In other words, one node on the road network may be represented with several vertices corresponding to the junctions, and they are independent with each other. Figure 1 gives the representation of different types of roads and junctions: one-way road, two-ways road without any access limitations, and T-junction with some access limitations (the center point on this T-junction is represented by two vertices in this directed graph).



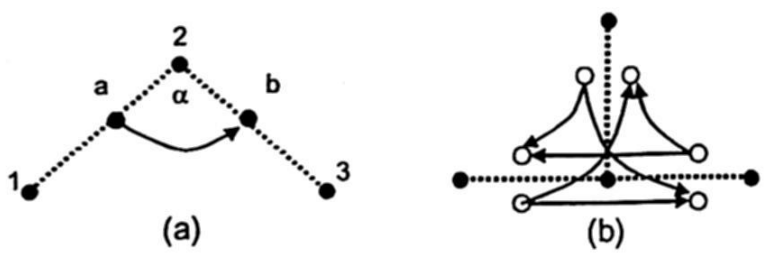


Figure 1. Different types of roads and junctions [7]

Figure 2. A primary graph (a, dotted lines) and its pseudo-dual graph (a, black), and a T-road primary graph (b, dotted lines) and its restricted pseudo-dual graph (b, black) [6]

Since this representation method ignores the spatial attributes of map objects, only the routing queries are applicable well on this model. An architecture was proposed in [8] for keeping traffic information on nodes of road network. However, the information of traffic constraints and turn costs on the nodes is omitted in their discussion. To represent the traffic cost and the cost of turns, a method in [6] was proposed. The turn cost is represented by a pseudo-dual graph. Figure 2 gives the pseudo-dual graph for representation of turn costs on a primary graph (Figure 2(a)) and a T-junction with some access limitations (Figure 2(b)). The turn cost is represented with the additional nodes and arcs of the pseudo-dual graph: in (a), two nodes *a* and *b* are added to the primary graph, and an arc α from *a* to *b* represents the turn cost of leaving the traffic arc from node *1* to node *2* and entering the traffic arc from node *2* to node *3*; and in (b), there are six nodes and six arcs which are added for representing turn costs between possible pairs of traffic arcs. The cost of search algorithms (e.g. Dijkstra's algorithm [9]) becomes high. Moreover, the pseudo-dual graph is insufficient (and needs reference to the primary graph) for route drawing.

In this paper, we propose a flexible method for representing transportation networks, including basic road map, traffic constraints, travel cost on road segments and (or) turn cost from a traffic link to other "turnable" links. Moreover, our method is proper for representing multi-level transportation network.

2.2 METHODS OF MANAGING MULTI-LEVELS OF TRANSPORTATION NETWORK

To achieve an efficient navigation process on transportation network, many methods were proposed to pre-compute hierarchical road networks [10, 11, 12, 13]. Proper partitions are done to the transportation network, and some parts of the path length are pre-computed. In their methods, all the road segments are regarded as the same level of detail - basically, all the navigation process is based on the most detailed map. They do not support the navigation based on transportation networks on different level of details.

Contrary to those methods based on the most detailed map, multi-resolution data models provide ways of describing the world at various levels of detail. The importance of such data models in the context of spatial information is widely acknowledged, and there are several studies of their formal foundations [14, 15, 16]. Under these models, to realize the sharability among multi-levels or rapid access to multi-levels of maps, map objects are arranged in multi-levels based on a compromise between storage and computation. Although the sizes of storage, the speed of map zooming or effective spatial process on multi-levels of maps were studied in these works, they share the same complex process to keep the consistency among multi-levels of map information. To solve this problem, an access method, on Multi-levels Object-Relation (MOR-tree) structure, which possesses the ability of the followings, is proposed [4]:

1. The ability of handling spatial data efficiently;
2. The ability to provide integrated access to multi-scale maps;
3. The ability of arranging the relations among multi-levels of maps.

MOR-tree was proposed based on M^2 (Multi-level/Multi-theme) map information model [17] for managing multi-levels (of scale) of maps. *M2* model is powerful to deal with various levels of maps uniformly, especially for managing multi-levels of road networks in comparison with other models [15, 16]. However, MOR-tree cannot be applied to transportation information directly, as the transportation information on road network is more complex than the original map. In this paper, an integrated representation method for multi-level transportation network is proposed.

3 REPRESENTATION OF TRANSPORTATION INFORMATION

Not only the kinds of information but also the management method of transportation information affects the processing efficiency of queries in ITS applications. In this section, we propose a representation method for integrating traffic information and spatial information about road network by considering the followings terms:

1. The traffic conditions change continuously, and the snapshot of conditions is recorded as traffic information. In comparison with the traffic information, the map of road network is seldom updated, and can be regarded as static information. Therefore, if the static information is managed by an efficient structure, the changes of traffic information associated with the road map should not disturb the stability of the structure.
2. The integrated representation should not only support the spatial query on road network and the temporal query on traffic information, but also support the interaction between these two kinds of queries.

A road network with nodes and links representing respectively the crosses and road segments can be regarded as an un-directed graph G, $G = (V, L)$, where V is a set of vertices $\{v1, v2, ...vn\}$, and L is a collection of lines $\{l1, l2, ...lm\}$. Traffic information on the road network is regarded as a directed graph G', $G' = (V, A)$, where V is a set of vertices $\{v1, v2, ...vn\}$, and A is a collection of arcs $\{a1, a2, ...ap\}$.

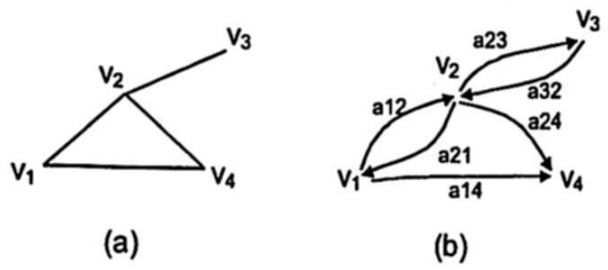


Figure 3. Road segment and traffic arc

Figure 3 depicts these two kinds of graphs. In the un-directed graph of Figure 3 (a) road segments are represented by lines, while in the directed graph of Figure 3 (b) junctions are represented by arcs. One line for road segment in Figure 3 (a) may be corresponded to two arcs in Figure 3 (b) with two-direction traffic information. In addition to the directions of traffic, there are usually traffic controls (constraints) on road network: for example, the right-turn and U-turn are forbidden on some cross-points, which constrain the action of traffic. The typical road junctions with (or without) constraints are given in Figure 4 and Figure 5. Road junctions are represented by using [7]'s model in Figure 4(a)(1), Figure 4(b)(1) and Figure 5(1). Considering the shortcomings of this simple model, we propose *super-node* representation method for integrating junctions (including traffic cost and traffic constraints) and road network.

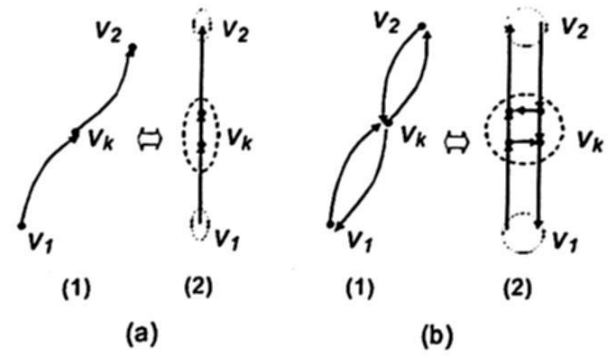


Figure 4. One-way road an d two-ways road

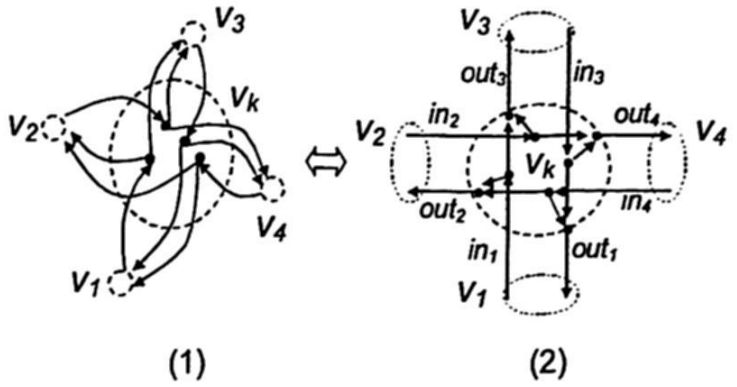


Figure 5. Cross node with constraint

A *super-node* can be defined as a node in road network: for example, *vk* in Figure 4 (a) (2), Figure 4 (b) (2) and Figure 5 (2). The information on the *super-node* contains the following parts (for simplicity of explanation, road junctions in Figure 5 (2) are used as an example):

1. *Cost-arc*: The arcs which have *vk* as their final vertex are called in-arcs, denoted as *ini*, and similarly the arcs which have *vk* as their initial vertex are called out-arcs, denoted as *outj*. The number of those arcs is called as in-degree (e.g., 4) and out-degree (e.g., 4), respectively. Every *outi* is defined as a *Cost-arc*, which consists of the final vertex of this arc and the traffic cost of this arc. *Cost-arcs* of v_k in Figure 5(2) are:

$$\begin{bmatrix} out_1(v_1, cost_{k1}) \\ out_2(v_2, cost_{k2}) \\ out_3(v_3, cost_{k3}) \\ out_4(v_4, cost_{k4}) \end{bmatrix}.$$

2. *Constraint-matrix*: The constraints on the *super-node* can be represented with an $n \times m$ matrix *CM*:

$$CM(v_k) = \begin{array}{c} \\ in_1 \\ in_2 \\ \vdots \\ in_n \end{array} \begin{array}{c} \begin{array}{cccc} out_1 & out_2 & \cdots & out_m \end{array} \\ \begin{pmatrix} C_{11} & C_{12} & \cdots & C_{1m} \\ C_{21} & C_{22} & \cdots & C_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ C_{n1} & C_{n2} & \cdots & C_{nm} \end{pmatrix} \end{array}$$

And C_{ij} equals to 1 when there is restriction from *ini* to *outj*; C_{ij} equals to 0 when there is a junction from *ini* to *outj*. *Constraint-matrix* for *vk* in Figure 5 (2) is:

$$CM(v_k) = \begin{array}{c} \\ in_1 \\ in_2 \\ in_3 \\ in_n \end{array} \begin{array}{c} \begin{array}{cccc} out_1 & out_2 & out_3 & out_m \end{array} \\ \begin{pmatrix} 1 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 \end{pmatrix} \end{array}$$

Where there are restrictions on going from *in1* to *out1* and *out4*, from *in2* to *out1* and *out2*, from *in3* to *out2* and *out3*, and from *in4* to *out3* and *out4*. If there is no restriction for any *ini* of the super-node *vk*, *Constraint-matrix* of *vk* is filled with *0*, and is regarded as *NULL*.

Moreover, our method is able to process the turn cost by extending *Constraint-matrix* to a *Turn-Cost/ Constraint-matrix*. The *CM* can be modified to a *Turn-Cost/Constraint-matrix*. For example, the *Turn-Cost/ Constraint-matrix* for *vk* in Figure 5 (2) may be like this:

$$T\_CM(v_k) = \begin{array}{c} \\ in_1 \\ in_2 \\ in_3 \\ in_n \end{array} \begin{array}{c} \begin{array}{cccc} out_1 & out_2 & out_3 & out_m \end{array} \\ \begin{pmatrix} MAX & 10 & 40 & MAX \\ MAX & MAX & 10 & 30 \\ 10 & MAX & MAX & 30 \\ 10 & 40 & MAX & MAX \end{pmatrix} \end{array}$$

Where *MAX* is defined as a large constant value. The element *TCij* in this matrix with a value of *MAX* represents a restriction from *ini* to *outj*: e.g., U-turn and right-turn are forbidden in this example. So, *MAX* is assigned to *TCii* (i = 1,..., 4), TC_{14}, TC_{21}, TC_{32} and TC_{43}. The value of TC_{12} represents the cost 10 (e.g., 10 seconds) of making a left-turn on the cross-point (from *in1* to *out2*), while the cost of crossing the point *vk* from *in1* to *out3* is 40.

This method decreases the redundancies in the database and it is easy to integrate the traffic information and the basic road network. For the basic road network, the additional information for traffic information is managed on every node. When the number of nodes and traffic arcs is unchanged, the modification to any of the traffic information does not injure the stability of the spatial index structure (e.g., R-tree [18]) for road network. Therefore, queries in ITS application, which refer to the spatial information, can be solved by taking advantages of the spatial index. Another kind of queries, which refer to traffic information, can also be solved effectively. In the next section, we center on solving the second kind of queries by integrating the transportation information with the multi-level road map.

4 REPRESENTATION METHOD OF MANAGING MULTI-LEVEL TRANSPORTATION NETWORK

Multi-level transportation network is an extended part of the multi-level road map model, and is managed by an extended MOR-tree.

4.1 M^2 MAP INFORMATION MODEL

M^2 map information model is designed for integrated management of multi-level/multi-theme maps. M^2 model can be regarded as a forest consisting of two kinds of hierarchies: one is a directory tree, which is obtained by recursively decomposing map regions into a sequence of increasingly finer tessellations with regard to the granularities of administrative units; and another is a theme tree, which is obtained by uniquely dividing spatial objects into different themes. In every theme, map objects are divided into different levels of scales in regard to the display needs of map [19]. The information model is powerful to integrate various scales of maps uniformly and possesses advanced extensibility [17].

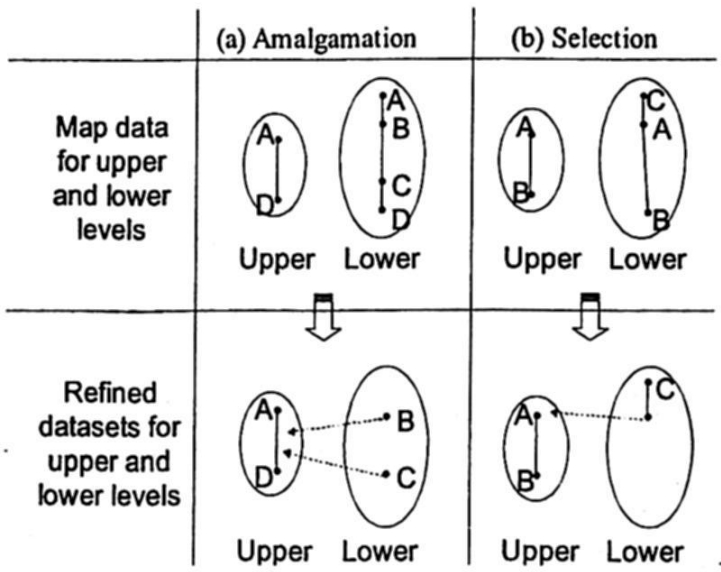


Figure 6. Refinement

Under M^2 model, road objects are assigned to multi-level datasets without repetition: the dataset in the lower level is a supplement of the dataset in the upper level. Typical situations of this assignment in two levels are given in Figure 6. In Figure 6(a), the relation between the map objects in upper and lower levels is amalgamation: the map object AD is represented as a line (AD) in the upper-level map and as lines (AB, BC, CD) in the lower level. An object in the upper level can be regarded as the combination of adjacent objects in the lower level. The information about nodes A and D is the same in both levels but the relation between A and D is different. So, under our model the information in the lower-level map can be split into two parts: one part is the same information as that in the upper level; and another part is the information, which belongs to only the lower level. These two parts are called *refined datasets*, and are managed in the upper and lower levels, respectively. Considering that the relations between two datasets are needed only in generating maps of the lower level, these relations are managed in the refined dataset of the lower level. In (b), there is a selection relation between maps in two levels: the map object AB is displayed in two levels, but AC is not needed in the upper level. The information in the upper level is a sub-set of that in the lower level.

Therefore, there are an object AB and an object AC in the refined datasets on upper and lower levels, respectively.

4.2 MOR-TREE FOR MULTI-LEVEL ROAD NETWORK

Besides the ability of accessing spatial objects in one level (in one dataset) efficiently just like other spatial index structures (e.g., R-tree [18]), MOR-tree is designed so as to be able to differentiate the levels of objects and manage the relations among objects of multiple levels by adopting two kinds of hierarchies. A main hierarchy is proposed to differentiate the levels of road objects by assigning logical importance values to every object. The logical importance value is a natural number in agreement with the map level: e.g., 1 for objects in country-level datasets, 2 for those in prefecture-level datasets, and so on. The objects with higher importance (with a smaller value) are stored in the higher levels of the hierarchy. Another kind of hierarchy, called a composition hierarchy, is introduced to keep the relations among levels, which are pointed by the leaf nodes of the main hierarchy. The main hierarchy is based on R-tree, to achieve the outstanding spatial access performance. Each node in MOR-tree contains a number of entries. There are three kinds of entries: tree-entries, object-entries and composition-entries. The internal nodes may contain the first two kinds of entries, in contrast to R-tree. The leaf nodes contain object-entries. The object-entry points to a composition hierarchy, which consists of composition-entries.

Three kinds of entries have the following forms:

1. Object-entry has the form (*MBR*, *flag*, *comp-ptr*), where *MBR* is the minimal bounding rectangle of composition hierarchy; *flag* is a natural number that indicates the importance level; and *comp-ptr* contains a reference to a composition hierarchy;
2. Tree-entry has the form (*MBR*, *flag*, *child-ptr*), where *child-ptr* contains a reference to a subtree. In this case *MBR* is the minimal bounding rectangle of the whole subtree and *flag* is the smallest importance value of the child-nodes.
3. Composition-entry has the form (*comp-id*, *n-ptr*, *nl-ptr*), where *comp-id* is the identifier of object's composition; *n-ptr* contains a reference to the next composition of the parent node object: e.g., the object is a road segment, its first composition is one of its end points, *n-ptr* points to the next point on the same object; and *nl-ptr* contains a reference to the composition of the parent node object in the lower level: e.g., the intersection between the upper-level road and lower-level road.

Here, we give object-entry and composition-entry in Figure 7 as an example. In Figure 7(a), there is a link object *a1* with compositions *na1* and *na2* in *level 1*, so the entry in the index structure for *a1* is an object-entry, which consists of *MBR* for *a1* (For simplicity, we use *a1* to replace *MBR* in this figure), *flag* of *level 1* (here it is *1*) and *comp-ptr* which points to one composition of *a1* (here it is *na1*). The compositions of *a1* (*na1*, *na2*) are managed by composition-entries. *n-ptr* of entry for *na1* points to the entry for *na2*, and the other pointers of two composition-entries point to NULL. A more complex example is given in Figure 7(b), in which there are lower level compositions (*nb1* and *nb3*) of *a1*. The objects *b1* and *b2* belong to *level 2*. So, *nl-ptr* of *na1* points to the composition-entry *nb1* of *level 2* and *n-ptr* of *nb1* points to *nb3*.

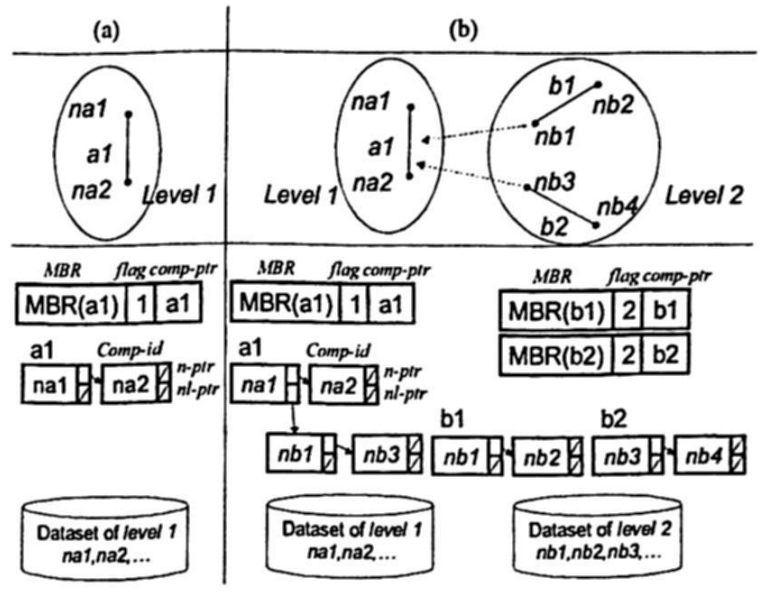


Figure 7. Example of MOR-tree for tw o-levels of road network

4.3 INTEGRATED REPRESENTATION OF MULTI-LEVEL TRANSPORTATION NETWORK

Basically, the transportation information is managed on every node of the map. However, the node, which connects to road segments in different levels, should be processed specially. Consider the multi-level transportation network given in Figure 8: *a1*, *a1* and *b1* to *b3* are basic road segments in *level 1* and *level 2*, respectively. The arcs (or dotted arcs) represent the transportation arcs in *level 1*(or *level 2*). For the simplicity of the figure, the traffic constraints are not drawn in this figure, but the constraint is set as U-turn forbidden, here. The MOR-tree for the basic road network in Figure 8 is given in Figure 9. The information about nodes is managed in *level 1* and *level 2* according to the level of basic map. The road segments between nodes of different levels can be recorded in the pointer of *nl-ptr* in the composition hierarchy. However, such pointers cannot assure that there are transportation arcs between these nodes: e.g., *na1* points to *nb1* and *nb2*, but there is no transportation arc from *na1* to *nb2* (there is only an arc from *nb2* to *na1* as *b3* is a one-way road). Consider that the path search in *level 1* is based on the arcs in *level 1* and the search in *level 2* is based on a different arc set.

For the search in *level 1*, *Cost-arcs* of node *a1* are:

$$\begin{bmatrix} out_{a1}(na_2, \cos t_1) \\ out_{a2}(na_3, \cos t_2) \end{bmatrix}.$$

While for that in *level 2* the *Cost-arcs* of node *a1* are:

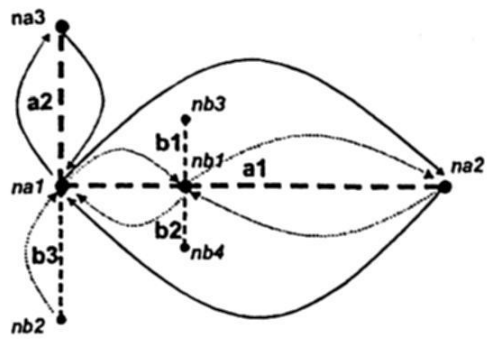


Figure 8. Multi-level trans portation network

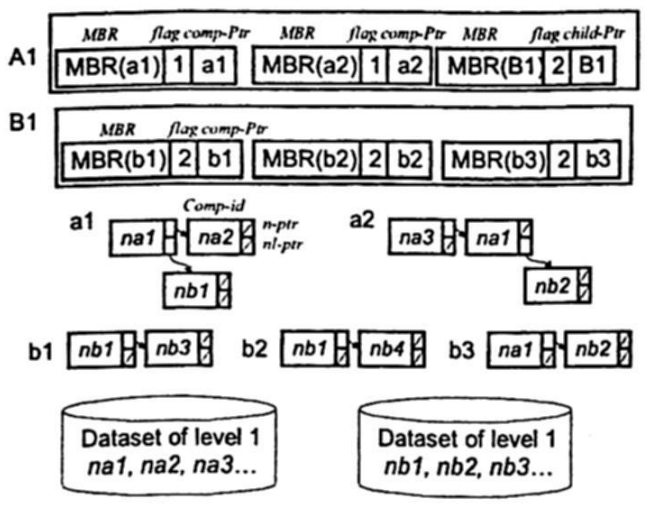


Figure 9. MOR-tree for multi-level road network: *Ai:* internal nodes of main hierarchy; *Bi:* leaf nodes of main hierarchy; a *i, bi:* composition hierarchy; nai, nbi*:* Composition-entries; *i (1 or 2):* flag of Object-entries or Tree-entries

$$\begin{bmatrix} out_{a11}(na_2, \cos t_{11}) \\ out_{a2}(na_3, \cos t_2) \end{bmatrix}.$$

The *Constraint-matrix* of node *a1* in *level 1* and *level 2* are:

$$CM(na_1)_1 = \begin{array}{c} \\ in_{a1} \\ in_{a2} \end{array} \begin{array}{c} \begin{array}{cc} out_{a1} & out_{a2} \end{array} \\ \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \end{array}$$

$$CM(na_1)_2 = \begin{array}{c} \\ in_{a11} \\ in_{a2} \\ in_{b2} \end{array} \begin{array}{c} \begin{array}{cc} out_{a11} & out_{a2} \end{array} \\ \begin{pmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{pmatrix} \end{array}.$$

Because 1) the information in the lower level (*level 2*) is not to be used by the operation in *level 1*; and 2) the original MOR-tree is not consistent with the transportation situation, we split one node in the upper level (*level 1*) into two nodes belonging to two levels: one is the original node in the upper level, and another is a transportation-node in the lower level, which contains transportation information of the lower level. In MOR-tree for road network, *nl-ptr* contains a reference to the composition of the parent node object in the lower level. Here, we redefine *nl-ptr* as a pointer, which contains a reference to a transportation-node of the parent node object in the lower level. The new tree is called as TMOR-tree, and its composition hierarchy is given in Figure 10.

The nodes *na1*, *na2* in *level 1* possess the transportation information referring to *level 2*, so new transportation-nodes for them are created in *level 2*. For the composition hierarchies of *level 1*, *nl-ptr*'s refer to the corresponding transportation-nodes in *level 2*. And, the search operations referring to only the upper level can be done just like by using the original MOR-tree, and the operations referring to the two levels can be done by using the information managed in all the levels. The nodes, which connect only with transportation arcs on the same level, can be represented with a *super-node* directly.

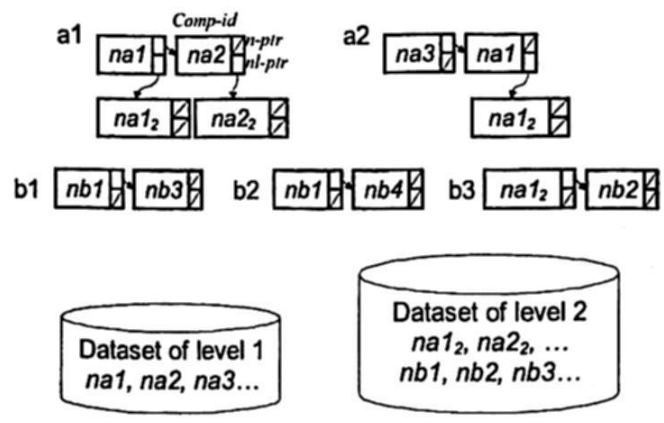


Figure 10. Composition nodes of TMOR-tree for transportation information

4.4 SPATIAL SEARCH AND PATH SEARCH

TMOR-tree supports not only the spatial search on multi-level maps but also the path search on multi-level transportation information. Spatial search to multi-level maps can be realized by accessing objects until a specific level via TMOR-tree. Since TMOR-tree takes the advantages of spatial index structure, spatial queries, such as region query, can be realized by accessing to the internal nodes and composition hierarchies until a specific level of TMOR-tree. The zoom-in/out operations can be realized by accessing the nodes under a specific internal node of TMOR-tree.

Path search on the transportation information on different levels can be realized by using an extended "ink-blot" search method. The ordinary "ink-blot" search method solves a path search from *vi* to *vt* likes this: it begins from expanding *vi*'s connecting nodes in the sequence of the *cost* on the *Cost-arc*; if the target node has not been expanded, the search goes on by expanding the nodes connected by those expanded nodes. For the search based on the multi-level transportation network, we extend the method. For example, a path search from city hall *A* to city hall *B* in separated prefectures can be realized by searching based on transportation networks on different level of details. The search can be done in three steps:

1. At first, the search is based on the city-level transportation network inside city *A* using "ink-blot"

search method. This step is terminated when a node belonged to the upper level is expanded.

2. Then, the search goes on in the upper level until a node inside the region city *B* is expanded.
3. At last, the search returns to city level and expands nodes until the target is found.

The correctness of this algorithm is assured by the "ink-blot" method. Here, we leave the proof out. By using TMOR-tree, the changeover points between multi-levels are those nodes on the road network, which possess original node and transportation-node on multiple levels.

5 ANALYSIS

In this section, we introduce our prototype system and compare our *super-node* representation method with the node-arc methods used by [6, 7].

Our prototype system was developed in Java on an SGI O2 R5000 SC 180 entry-level desktop workstation. The system manages basic road maps in a part of Aichi Prefecture, Japan. The map is divided into map pages with the same size of *2000m × 1500m* (which is defined by Geographical Survey Institute of Japan [20]. As it has no effectiveness on the proposed methods in this paper, we ignore this concept in the followings). The total number of nodes *Nnum* is 42,062 and the number of links *Lnum* is 60,349 in the basic road map. These road segments are assigned to the country level (country-wide highways and national roads), the prefecture level (prefectural roads and main local roads) and the city level (city roads).

In city level transportation network, the average traffic arcs connecting to a node is about *2.87* (*=2 Lnum / Nnum*). When there is no traffic constraint for the basic road map, in node-arc method [7], there are 120,798 records (two times of the link numbers in road maps). As in our *super-node* method, the amount of information is related to the number of arcs of every node: here, the nodes with four, three, two and one out-arcs are about 24: 51: 13: 12. The total arcs managed in city level by SN method are 120,798. When there are traffic constraints, Right-turn and U-turn are forbidden in about half of the cross and T-junction points. Then, in NL method there are about 142,423 nodes and 135,293 arcs; while in SN method the amount of information keeps the same whether there are constraints (turn costs) or not. The number of arcs and nodes in city level (country and prefecture level) managed by *super-node* method (denoted as SN) and those by node-arc method (denoted as NA) are given in Figure 11 (Figure 12). In the figures, there are different values for different conditions of datasets in NA method. "Constraint" means there are traffic constraints and "Turn" means there are turn costs in the dataset. Because the number of nodes and arcs keep the same without depending on there are constraints (turn costs) or not, the datasets generated by SN method shares the same value in this figure, which is denoted simply as SN_method.

We have done experiments of finding nearest target object based on these datasets. The test results are given in Figure 13. In the figure, x-axis represents the density of targets on the road network, which is the ratio of the targets' number (*Tnum*) to the nodes' number (*Nnum*) in basic road map; and y-axis represents the CPU time for the search. D represents that adopting Dijkstra's algorithm does the search; and R represents the extended "ink-blot" search method proposed in this paper. D algorithm is executed based on the datasets generated by NA method with/without constraint (the number of objects is depicted in Figure 11); and R algorithm is done on the dataset generated by SN method with/without constraint. In Figure 15, we can observe that our SN_R methods outperform those of NA_D at any situations: especially, when there are traffic constraints on the networks.

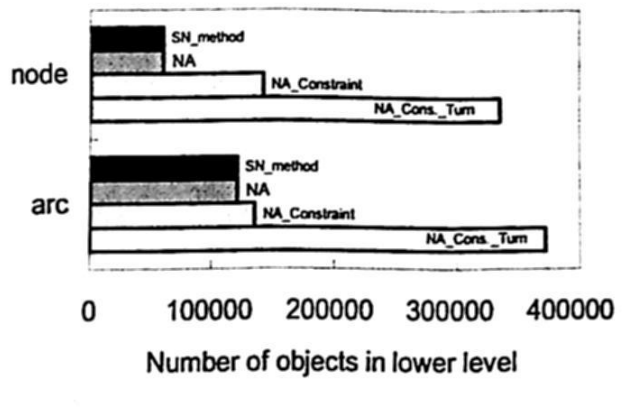


Figure 11. Numbers of arcs and nodes in city level managed by super-node and node-arc methods

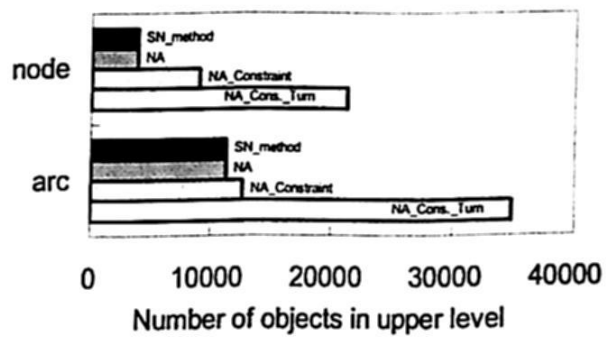


Figure 12. Numbers of arcs and nodes in country and prefecture level managed by super-node and node-arc methods

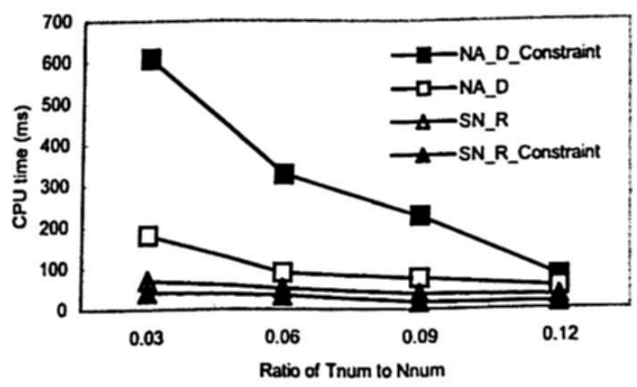


Figure 13. CPU time of finding nearest target object

6. CONCLUSIONS

In this paper, we proposed a representation method for multi-level transportation networks. Our method adopts a spatial index structure for managing map objects in multiple levels and uses an integrated method for representing the travel junctions (or traffic constraints), travel cost on road segments and turn corners. Based on the datasets generated by this method, queries in ITS applications can be responded efficiently in different levels of details. In our future work, the performance of the creation, modification and processing of the datasets created by our method will be evaluated, deeply.

REFERENCES

[1] S. Shekhar and D.R. Liu, CCAM: A connectivity-clustered access method for aggregate queries on transportation networks: a summary of results, *Proc. of ICDE'95*, 1995, 410-419.

[2] B. Becker, H.W. Six and P. Widmayer, Spatial priority search: an access technique for scaleless maps, *Proc. of ACM SIGMOD'91*, 1991, 128-137.

[3] P.V. Oosterom, The Reactive-tree: a storage structure for a seamless geographic database, *Proc. of Auto-Carto, Vol. 10*, 1991, 393-407.

[4] J. Feng and T. Watanabe, MOR-tree: An access structure for multi-levels of road networks on distributed environment, *Journal of the ISCIE, Vol.16, No.12,* 2003, (to appear).

[5] M. F. Goodchild, GIS and transportation: status and challenges, *GeoInformatica, Vol.4, No.2*, 2000, 127-139.

[6] S. Winter, Modelling costs of turns in route planning, *GeoInformatica, No. 4*, 2002, 345-361.

[7] J. Fawcett and P. Robinson, Adaptive routing for road traffic, *IEEE Computer Graphics and Applications, Vol. 20, No.3*, 2000, 46-53.

[8] D. Papadias, J. Zhang, N. Mamoulis and Y.F. Tao, Query processing in spatial network databases, *Proc. of VLDB 2003*, 2003, 802-813.

[9] N. Christofides, *graph theory: an algorithmic approach*, Academic Press Inc.(London) Ltd. 1975.

[10] N. Jing, Y.W. Huang and E.A. Rundenstteiner, Hierarchical encoded path views for path query processing: an optimal model and its performance evaluation, *IEEE Transactions on Knowledge and Data Engineering, Vol. 10, No. 3, 1998,* 409-431.

[11] Y.W. Huang, N. Jing and E.A. Rundensteiner, effective graph clustering for path queries in digital map, *Proc.of CIKM'96*, 1996, 215-222.

[12] Y.W. Huang, N. Jing and E.A. Rundensteiner, Path queries for transportation networks: dynamic recordering and sliding window paging techniques, *Proc. of GIS'96*, 1996, 9-16.

[13] E.P.F. Chan and N. Zhang, Finding shortest paths in large network systems, *Proc. of GIS'01*, 2001, 160-166.

[14] J. G. Stell, Granulation for graphs, *Lecture Notes in Computer Science, Springer-Verlag, No. 1661*, 1999, 417-432.

[15] Y. Leung, K.S. Leung and J.Z. He, A generic concept-based object-oriented geographical information system, *Int'l J. of Geographical Information Science, Vol.13, No. 5*, 1999, 475-498.

[16] S. Timpf, Hierarchical structures in map series, 1998, on-line: http://www.geoinfo.tuwien.ac.at

[17] J. Feng and T. Watanabe, Effective representation of road network on concept of object orientation, *Trans. IEE of Japan, Vol. 122-C, No. 12,2002,* 2100-2108.

[18] A. Guttman, R-Trees: A dynamic index structure for spatial searching, *Proc. of ACM SIGMOD'84*, 1984, 47-57.

[19] T. Yamanashi and T. Watanabe, Multi-layers/Multi-phases model adaptable to integrated GIS, *Proc. of VSMM'00*, 2000, 668-676.

[20] http://www.gsi.go.jp/MAP/CDROM2500/t2500.htm