STUDY ON THE CHANGE DETECTION MODEL AND METHOD FOR NAVIGATIONAL DIGITAL MAP DATA

Lianying Li [a,b] *, Lifan Fei [a]

[a] School of Resource and Environment Science, Wuhan University，Luoyu Road 129, Wuhan, Hubei, China
[b] State Key Laboratory of Information Engineering in Surveying; Mapping and Remote Sensing, Wuhan University，Luoyu Road 129, Wuhan, Hubei, China

Commission VI, WG II/1

KEY WORDS: Spatial Analysis, Detection, Computer Vision ,Spatio-temporal Modeling, Navigation

ABSTRACT:

A necessary updating degree is vital for the electronic map data in a vehicle navigation system. Only when the digital map data are well updated, can the quality of the navigation be assured. Today the companies devoting to the production of electronic map data for vehicle navigation have to cost much labor, material and capital to collect and update data in order to maintain a suitable updating degree. Ordinarily the updating frequency of the mother database in the server is a quarter or half of a year.This paper focuses on the automatic change detection of data that is one of the key techniques for incremental updating. In our research, with the basic principles of the Spatio-temporal data model, and considering the characteristics of data for navigational electronic map, we have established a multi-level rule set for detection of changed elements, and from the perspectives of geometry, attributes, and topology, expressed the changes formally with rules. Further more, we have designed the experiment for change detection; it has proved that the correctness from the automatic detection is just the same with that from the manual work.

1. INTRODUCTION

As a high-tech product which strongly integrates software and data, GPS Traffic Navigation System demands high interdependence and restriction on its software and data. Excellent software can actually serve for the users only by the data of high-quality and high-updating-degree. However, what is the situation nowadays? In the world, most of soft wares are high maturity, as well as timely updating. On the contrary, the related data is always unalterable, and updates infrequently. For instance, data manufacturers in home are able to update data twice generally, and the ones at abroad can at most afford three or four times per year currently. This problem always leads navigation failures, which perplex the users. How to make better use of the updating-degree, accuracy and information content of navigation electronic map - the foundational and critical part of traffic navigation system - has been the dilemma in the field of navigation system development in the whole world. In fact, those experts has research it from data model and updating methods [FAN Da-zhao, 2005; Flecher, D.,1987; Gottsegen J.,1994; Jun Feng,2005a].

In China, corresponding surveying and mapping departments, as well as traffic navigation enterprises have accumulated amounts of navigation electronic map through many years effort, in fact these data were stored in the multi-version database, generally each version of the data are independent. Now,there are two questions here. One of the questions is how to organize and manage massive data efficiently, the other is how to find the difference between these multi-version navigational data? The key to solve the two problems is “Change Detection”.

This paper focuses on the automatic change detecton of navigation electronic map, that is one of the key techniques for data organization, multi-version relations and incremental updating. those experts has research the change algorithm [Deren LI,2002; U. C. HERZFELD,1988; EUGENE W. MYERS,1992; Merriam, D. F.,1988],In our research, with the basic principles of the Spatio-temporal data model, and considering the characteristics of data for navigational electronic map, we have established a multi-level rule set for detection of changed elements, and from the perspectives of geometry, attributes, and topology, expressed the changes formally with rules.

2. DEFINITION OF NAVIGATIONAL DATA CHANGES

The model of navigational data is the foundation not only for effective organizing, accessing and managing the digital map data, but also for effective data transmission, exchange and application. Now, navigational data contain background, road, POI (Points of interest) and guidance data. To detect these changes, we divide these changes of the features into geometric change (Ch_{Geo}), attribute change (Ch_{Attr}) and topology change (Ch_{Topo}) , and design some methods to detect every kinds of these changes.

2.1 Detection of geometric change

In the process of Spatio-temporal evolution, the road can be expressed formally, though the shape changes. For example, the coordinates of the road nodes are changed, the number of shape points in the road arc is changed and some coordinate is changed in the road arc.

* Corresponding author: lily@whu.edu.cn

Definition of geometric shape status set

Define set S_{Geo}={Geo_1, Geo_2, ⋯ Geo_i, ⋯ Geo_n}. Where Geo_i represents the geometric shape of the road i; S_{Geo} represents a geometric shape status set which contains n road data in the navigational electronic map.

Define set S(t,Geo) ={<t,Geo>|t⩾0 且 t∈R,Geo∈SGeo }. Where S(t,Geo) represents the status of the data in the navigational electronic map at the time t.

Define $S_{GeoNode}$={<$NodeID_1$,x_1,y_1>, <$NodeID_2$,x_2,y_2>,… <$NodeID_i$,x_i,y_i>,… <$NodeID_n$,x_n,y_n>,} set to represent the geometric location of the road nodes, and

S_{GeoArc}={<$ArcID_1$,n_1,x_{11},y_{11},…x_{1n_1},y_{1n_1}>,
<$ArcID_2$,n_2,x_{21},y_{21},…x_{2n_2},y_{2n_2}>,…
<$ArcID_i$,n_i,x_{i1},y_{i1},…x_{ini},y_{ini}>,…
<$ArcID_n$,n_n,x_{n1},y_{n1},…x_{nn_n},y_{nn_n}>} to represent the geometric shape of the road arcs.

Geometric change operators

Define (t_1,<$NodeID_1$,x_1,y_1>) as the geometric shape of the road data at the time of t_1, and (t_2,<$NodeID_2$,x_2,y_2>) as the geometric shape of the road data at the time of t_2. If $NodeID_1$=$NodeID_2$, while $x_1 \neq x_2$ or $y_1 \neq y_2$, we can get a conclusion that there is a geometric change on the data of road nodes, and then we mark it as $Ch_{GeoNode}$ = True.

Define (t_1, <$ArcID_1$, n_1, $x_{1,1}$, $y_{1,1}$, ⋯ $x_{1,n1}$, $y_{1,n1}$>) as the geometric shape of the road data at the time of t_1, and (t_2, <$ArcID_2$, n_2, $x_{2,1}$, $y_{2,1}$, ⋯ $x_{2,n2}$, $y_{2,n2}$>) as the geometric shape of the road data at the time of t_2. If $ArcID_1$=$ArcID_2$ and $n_1 \neq n_2$, we can get a conclusion that there is a geometric change on the data of road arcs, and mark it as Ch_{GeoArc_n} = True . If $ArcID_1$=$ArcID_2$ and n_1=n_2, while $x_{1,i} \neq x_{2,i}$ or $y_{1,i} \neq y_{2,i}$ (i>1）, we can get a conclusion that there is a geometric change on the data of road arcs, and then mark it as Ch_{GeoArc_xy} = True . We mark geometric change of the road shape as Ch_{GeoArc} = True.

Algorithm design of Geometric change detection

Before change detection, we have to synchronize their feature IDs, so that the data of features in the old and new versions can be referenced and compared with each other quickly. Then based on the definitions and operators of data change, we design an algorithm to detect the geometric changes.

Geometric change algorithm

According to Geometric change operators, we can design an algorithm as followed, the algorithm named "traversal algorithm":

(1) Import the new and old version data into database (Oracle 10G);
(2) Determine whether it is the end of the database, if so, then go to (step 7), else go to (step 3);
(3) Fetch road feature ID and its shape points' coordinates from the old version map database;
(4) Search a road feature with same ID in the new version map database, and query its corresponding coordinates;
(5) If not found the feature with the same ID , or the number of shape points is not equal, or some point's coordinate is not equal, then there must be a geometric change between the two version's map database;
(6) Go to (step 2), continue to detect the next road feature's geometric change;
(7) Exit.

The flow chart of this algorithm is shown in figure1.
FID_new _i_n1 represents that the arc has n1 points in the new version database;
FID_new_i_Coordinate1() represents each point's value in the arc in the new version database;
FID_old _i_n2 represents that the arc has n2 points in the old version database;
FID_new_i_Coordinate2() represents each point's value in the arc in the old version database;

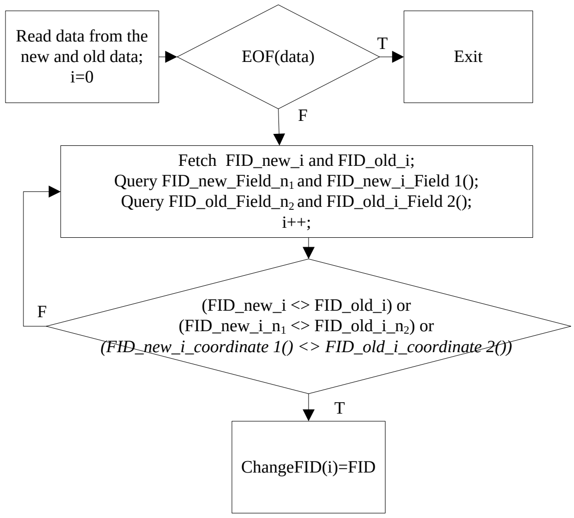

Figure 1. The flowchart of geometric change detection

When we did the change detection experiment, we found that the algorithm was not efficient because the process of comparing coordinate values cost too much time. As we all known that Image Change Detection Algorithms are mature and efficient, Thus, we redesigned the algorithm based on image. The algorithm named "rasteration method" includes four steps as followed.

(1) Vector to raster conversion: In the process, only the mid-point of segment is recorded as pixel, the built image is not complete raster image in order to save computer memory resources;

(2) Statistical Data computing: to count the points number according to the line and column respectively in new (or old) version map and write it in file named "new" (or "old");

(3) Statistical Data comparing: after finished step 2, we should compare the total points number each line(column) between "new" and "old" file , if the number is different, the line is changed and we should continue to search which pixel is changed;

(4) Pixel to coordinate conversion: translate the changed pixel's line number and column number into coordinates, and then the feature in vector map is easily found.

The process of rasteration method as shown in figure 2 .

Experimental Area	Segment Number of Old version Map	Segment Number of new version Map	"rasteration method" time cost (ms)	"traversal algorithm" time cost (ms)
10*10 km^2	1306	1681	1	8
25*25 km^2	2987	3589	2	22
50*50km^2	8467	8850	4	55
100*100km^2	28985	29629	16	225
150*150km^2	71051	73077	39	830
200*200km^2	131028	138024	81	3311
250*250km^2	210139	240173	170	23171
300*300km^2	282815	321468	234	38644
350*350km^2	319126	358436	261	44621
400*400km^2	374728	415110	302	53137

Table 1. The efficiency comparison of two algorithms

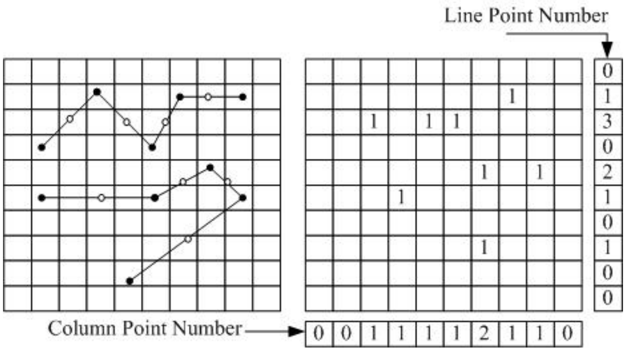

Figure 2. The process of rasteration method

In order to verify the efficiency of two algorithms, we use different sizes of the experimental zone for a comparative experiment, the results as shown in the table 1. This shows that: With the expansion of the experimental area, the "rasteration method" is more efficient than the "traversal algorithm".

2.2 Detection of attribute change

There are varied attribute information in the navigation map database, i.e. road name, road width and other attribute maybe change. Before detecting these changes we should express them formally.

Definition of attribute status set

Define set $S_{Attr}=\{Attr_1, Attr_2,\ldots Attr_i,\ldots Atrt_n\}$。Where $Attr_i$ represents the attribute of the road i; S_{Attr} represents a attribute status set which contains n road data in the navigational electronic map.

Define set $S_{(t,Attr)}=\{<t,Attr>| t\geqslant 0$ 且 $t\in R,\ Attr\in S_{Attr}\}$. Where $S_{(t,Attr)}$ represents the status of the data in the navigational electronic map at the time t.

Define $S_{AttrNode}=\{<NodeID_1,n1, Attr_{11}, Attr_{12},\ldots,Attr_{1n1}>,<NodeID_2,n2, Attr_{21}, Attr_{22},\ldots,Attr_{2n2}>,\ldots<NodeID_i,ni, Attr_{i1}, Attr_{i2},\ldots,Attr_{ini}>,\ldots,<NodeID_n,nn, Attr_{n1}, Attr_{n2},\ldots,Attr_{nnn}>\}$ set to represent the attribute of the all kinds of road nodes, and

$S_{AttrArc}=\{<ArcID_1,n1, Attr_{11}, Attr_{12},\ldots,Attr_{1n1}>, <ArcID_2,n2, Attr_{21}, Attr_{22},\ldots,Attr_{2n2}>,\ldots <ArcID_i,ni, Attr_{i1}, Attr_{i2},\ldots,Attr_{ini}>,\ldots, <ArcID_n,nn, Attr_{n1}, Attr_{n2},\ldots,Attr_{nnn}>\}$ to represent the Attributes of the all kinds of road arcs.

Attribute change operators

Define $(t_1, <NodeID_1,n1, Attr_{11}, Attr_{12},\ldots,Attr_{1n1}>)$ as the attribute of the road nodes at the time of t_1, and $(t_2, <NodeID_2, n2, Attr_{11}, Attr_{12},\ldots,Attr_{1n2})$ as the attribute of the road nodes at the time of t_2. If $NodeID_1=NodeID_2$, while $n1\neq n2$ or $Attr_{1i}\neq Attr_{2i}$, we can get a conclusion that there is a attribute change on the data of road nodes, and then we mark it as $Ch_{AttrNode}$ = *True*.

Define $(t_1, <ArcID_1, n1, Attr_{11}, Attr_{12},\ldots,Attr_{1n1}>)$ as the attribute of the road arc at the time of t_1, and $(t_2, <ArcID_2, n2, Attr_{11}, Attr_{12},\ldots,Attr_{1n2}>)$ as the attributes of the road arc at the time of t_2. If $ArcID_1=ArcID_2$ and $n1\neq n2$, we can get a conclusion that there is a attribute change on the data of road arcs, and mark it as $Ch_{AttrArc\_n}$= *True* . If $ArcID_1=ArcID_2$ and $n1=n2$, while $Attr_{1i}\neq Attr_{2i}$ $(i>1)$, we can get a conclusion that there is a attribute change on the data of road arcs, and then mark it as $Ch_{AttrArc\_Field}$ = *True* . We mark attribute change of the road attribute as $Ch_{AttrArc}$ = *True*.

Attribute change algorithm

(1) Import the new and old version data into database (Oracle 10G);
(2) Determine whether it is the end of the database, if so, then go to (No. 7 step), else go to (No. 3 step);
(3) Fetch road feature ID and its attribute information from the old version map database;
(4) Search the road feature with same ID in the new version map database, and query its corresponding attribute information;
(5) Comparing the corresponding attribute information between two versions, once some attribute field value is different, mark it with "attribute change";
(6) Go to (No. 2 step), continue to detect next road feature's attribute change;
(7) Exit.

The flow chart of this algorithm is shown in figure3.

FID_new _Field_n1 represents that the arc has n1 fields in the new version database;

FID_new _Field1() represents each field's value in the arc in the new version database;
FID_new _Field_n2 represents that the arc has n2 fields in the new version database;
FID_new _Field2() represents each field's value in the arc in the old version database;

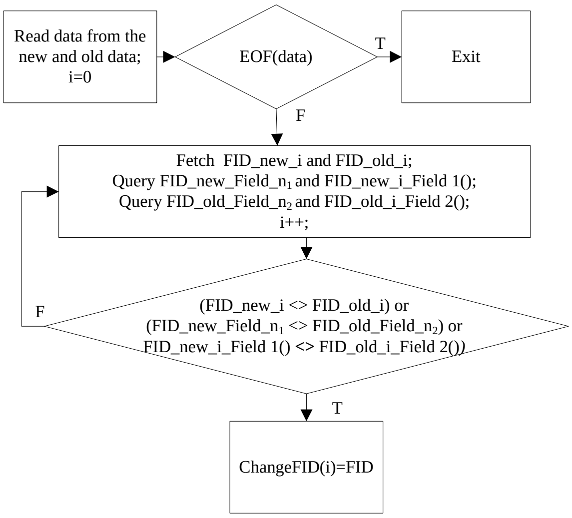

Figure 3. The flowchart of attribute change detection

2.3 Topology change detection

Topology change detection includes arc-node topology and node-arc topology. Arc-node topology represents the relationship between the arc and the start & end node; node-arc topology represents how many arcs connect with the node.

Definition of Topology status set

Define set $S_{Topo}=\{Topo_1, Topo_2,\ldots Topo_i,\ldots Topo_n\}$。 Where $Topo_i$ represents the Topology of the road i; S_{Topo} represents a topology status set which contains n road data in the navigational electronic map.

Define set $S_{(t,Topo)}=\{<t,Topo>|\, t\geqslant 0$ 且 $t\in R,\ Topo\in S_{Topo}\}$. Where $S_{(t,Topo)}$ represents the status of the data in the navigational electronic map at the time t.

Define

$S_{ToporNode}=\{<NodeID_1,n1,\ ArcID_{11},\ ArcID_{12},\ldots,\ ArcID_{1n1}>,$
$<NodeID_2,n2,\ ArcID_{21},\ ArcID_{22},\ldots,\ ArcID_{2n2}>,$
$\ldots$
$<NodeID_i,ni,\ ArcID_{i1},\ ArcID\ r_{i2},\ldots,\ ArcID_{ini}>,$
$\ldots,$
$<NodeID_n,nn,\ ArcID_{n1},\ ArcID_{n2},\ldots,\ ArcID_{nnn}>\}$
set to represent the topology of the all kinds of road nodes, and

$S_{AttrArc}=\{<ArcID_1,sNodeID_{11},eNodeID_{12}>,<ArcID_2,sNodeID_{21},eNodeID_{22}>\ldots$
$<ArcID_i,sNodeID_{i1},eNodeID_{i2}>\ldots$
$<ArcID_n,sNodeID_{n1},eNodeID_{n2}>\}$ to represent the Attributes of the all kinds of road arcs.

Topylogy change operators

Define $(t_1,\ <NodeID_1,n1,\ ArcID_{11},\ ArcID_{12},\ldots,\ ArcID_{1n1}>)$ as the topology of the road nodes at the time of t_1, and $(t_2,\ <NodeID_2,\ n2,\ ArcID_{21},\ ArcID_{22},\ldots,\ ArcID_{2n2}>)$ as the topology of the road nodes at the time of t_2. If $NodeID_1=NodeID_2$, while $n1\neq n2$ or $ArcID_{1i}\neq ArcID_{2i}$, we can get a conclusion that there is a Topology change on the data of road nodes, and then we mark it as $Ch_{TopoNode}=True$.

Define $(t_1,\ <ArcID_1,\ sNodeID_{11},eNodeID_{12}>)$ as the topology of the road arc at the time of t_1, and $(t_2,\ <ArcID_2,\ sNodeID_{21},eNodeID_{22}>)$ as the topology of the road arc at the time of t_2. If $ArcID_1=ArcID_2$ and $(eNodeID_{112}\neq eNodeID_{21}$ or $eNodeID_{12}\neq eNodeID_{22})$, we can get a conclusion that there is a topology change on the data of road arcs, and mark it as $Ch_{TopoArc}=True$.

Attribute change algorithm

(1) Import the new and old version data into database (Oracle 10G);
(2) Determine whether it is the end of the database, if so, then go to (No. 7 step), else go to (No. 3 step);
(3) Fetch road feature ID and its Topology information from the old version map database;
(4) Search the road feature with same ID in the new version map database, and query its corresponding Topology information;
(5) Comparing the corresponding Topology information between two versions, once some Topology information is different, mark it with "Topology change";
(6) Go to (No. 2 step), continue to detect next road feature's Topology change;
(7) Exit.

The flow chart of this algorithm is shown in figure 4.
FID_new_Node_n1(FID_old_Node_n2) represents how many arcs connect with the node in the new(old) version database;
FID_new_i_Arc1()(FID_new_i_Arc1()) represents each ArcID connected with the node in the new(old) version database;
FID_new_Arc_i_sNodeID(FID_old_Arc_i_sNodeID) reprensents the arc's start node in the new(old) version database;
FID_new_Arc_i_eNodeID(FID_new_Arc_i_eNodeID) reprensents the arc's end node in the new(old) version database;

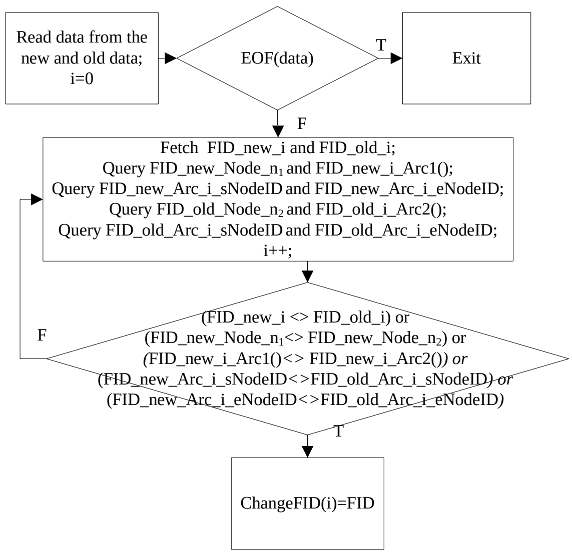

Figure 4. The flowchart of topology change detection

The algorithms for attribute & topology change detections are similar to that of geometry change detection.
The algorithm of change detection has been finished till now, but it is important to note that these kinds of change influence each other. If the road shape has changed, its attribute and topology maybe changed too. Consequently, when we research

the change detection in the navigation map database, we firstly find whether the geometric change is detected, if so then we should continue to search next feature rather than to detect the attribute change or topology change.
We have done an experiment to verify the validity of this method. Figure 5 describes respectively geometry, attributes, topology change. The road in black and bold style is the data of old version; on the other hand, the grey and thin style is the data of old version.

Figure 5. Geometric change

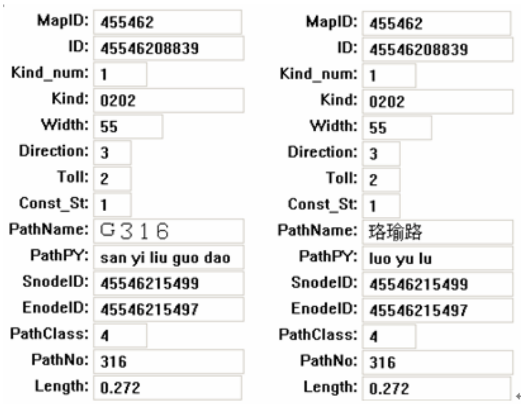

Figure 6. Attribute change

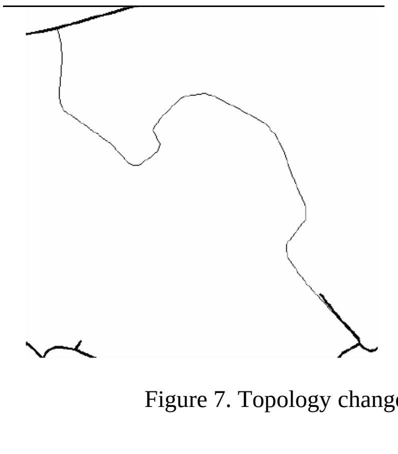

Figure 7. Topology change

The experiment has proved that the accuracy of the automatic detection is just the same as that of manual work. The substitution of the heavy, tedious and time-consuming manual work of change detection by the computers can not only greatly reduce the workload, substantially save the time, but also reliably strengthen the accuracy of the change detection and data updating.

3. EXPERIMENT AND CONCLUSION

This paper focuses on the automatic change detection of data that is one of the key techniques for incremental updating. In our research, with the basic principles of the Spatio-temporal data model, and considering the characteristics of data for navigational electronic map, we have established a multi-level rule set for detection of changed elements, and from the perspectives of geometry, attributes, and topology, expressed the changes formally with rules. Further more, because of time cost in traversal algorithm, we have designed a efficient "rasteration method" based on image change detection. We have done an experiment with navigational digital map data of Wuhan, versions 2006 and 2007, to verify the validity of this method. The experiment has proved that the correctness from the automatic detection is just the same with that from the manual work.

REFERENCES

FAN Da-zhao, ZHANG yong-sheng, LEI Rong, GUO Hai-tao,2005, Research on automatic updating of GIS data, Science of Surveying and Mapping, 30(3): 15-17(Chinese)

HAN Wei-guo, WANG Jin-feng, GAO Yi-geHU, Jian-jun,2007, Forecasting and Analysis of Regional Traffic Flow in Space and Time, Journal of Highway and Transportation Research and Development,6:92-96. (Chinese)

Deren LI , Haigang SUI & Ping XIAO. Automatic Change Detection of Geo2spatial Data from Imagery[A]. In : ISPRS , 2002 ,Xi'an[C] ,2002 :2452251.

U. C. HERZFELD and M. A. SONDERGARD. MAPCOMP--A FORTRAN PROGRAM FOR WEIGHTED THEMATIC MAP COMPARISON[J]. Computer and Geoscience Vol 14. No 5. pp. 699-713. 1988

EUGENE W. MYERS, XIAOQIU HUANGt. AN O(N 2 log N) RESTRICTION MAP COMPARISON AND SEARCH ALGORITHM[J]. Bulletin of Mathematical Biology Vol. 54, No. 4, pp. 599~618, 1992.

Merriam, D. F. and Jewett. D. G., 1988, Methods of thematic map comparison, in Current trends in geomathematics: Plenum Press. New York. in press.

ZENG Wen, YIN Yi, 2002, THE PRELIMINARY RESEARCH ON TEMPORAL SPATIAL PREDICTIVE MODEL OF URBANIZATION, Economic Geography,2:128-131. (Chinese)

Andreas Hecht, 2005,The Future Direction of Map Formats In Map-Enabled Applications,12th World Congresses ITS,6-10 November 2005, San Francisco, USA.

Atsushi SAKAMOTO, Yutaka OHSAWA, Shirou AKAI, 2000, A Proposal on Data Renewal Method for Car Navigation Systems, Proc.of ISPRS2000,CD-ROM Proceeding.

Bernhard Lorenz, 2005, Ontology of Transportation Networks, http://rewerse.net/deliverables/m18/a1-d4.pdf

Buttenfield B.P., 1995, Object-Oriented Map Generalization: Modeling and Cartographic Considerations, In: MullerJ. C. etal (eds),GIS and Generalization-Methodology and Practice,Taylor&France,91-105.

Car A,Frank AU.,1994,General Principles of Hierarchical Spatial Reasoning- The Case of Way finding, Proceedings of the Conference on Spatial Data Handling(SDH94),Edinburgh,Scotland,2:646-664.

Dueker K.J.,Butler J.A.,1998,GIS-T Enterprise Data Model with Suggested Implementation Choices, URISA Journal, 10(1):12-36.

Egil Juliussen, 2004, Telematics: Status & Future Perspectives, http://www.telematicsresearch.com/PDFs/TRG_ITSWG-Telematics.pdf.

Flecher, D., 1987, Modeling GIS Transportation Networks, In Proceedings, 25th annual meeting of the Urban and Regional Information System Association, 84-92.

Goodchild, M. F., 1998, The Relationship Between Geographic Information Systems and Disaggregate Transportation modeling, Geographic Systems, 5: 9-17.

Goodchild, M. F., 2000, GIS and Transportation: Status and Challenges, GeoInformatica 132 4(2):127-139.

Gottsegen J., Goodchild, M. F., and Church R., 1994, A Conceptual Navigable Database Model for Intelligent Vehicle Highway Systems. Proceedings, GIS/LIS Conference, Phoenix, AZ.

ISO/TC204, 2000, PSF and API Standard – Part 1: Application Categories and Logical Data Model, Issued 20.01.

ISO/TC204/WG3, 2004, Intelligent Transport Systems -- Geographic Data Files (GDF) Overall Data Specification.

Jiang J., Hang G., Chen J., 2002， Modeling Turning Restrictions in Traffic Network for Vehicle Navigation System, In Proceedings of the International Symposium on Geospatial. Theory, Processing and Applications Commission IV, Ottawa, 2002.

Jing N., Huang Y. W., Rundensteiner E. A., 1998, Hierarchical Encoded Path Views for Path Query Processing: An Optimal Model and Its Performance Evaluation, IEEE Transactions on Knowledge and Data Engineering, 10(3): 409～431.

Jun Feng, Yuelong Zhu, Naoto Mukai, et al, 2005a, Integrated Management of Multi-level Road Network and Transportation Networks, Khosla R. et al. (Eds.): KES 2005, LNAI 3683, pp. 677–683, Springer-Verlag Berlin Heidelberg.

Liang C., Lee C.H., Lee J.D., et al, 2001, Scale-Dependent Transmission of Spatial Vector Data on the Internet, In:The 3rd International Conference on Information Integration and Web-based Applications & Services(Austria).

Longley Paul A., Goodchild M. F., Maguire D. J., et al, 1998, Geographic Information Systems, Volume 1, Principles and Technical Issues, Second Edition, John Wiley & Sons Inc.

Mark D. M., Lauzon J. P., and Cebrian J. A., 1989, A Review of Quadtree-Based Strategies for Interfacing Coverage Data with Digital Elevation Models in Grid Form, International Journal of Geographical Information Systems, 3(1):3- 14.

OGC, 2003, OpenGIS Location Services (OpenLS): Core Services, OGC Implementation Specification, OGC 03-006r1, Open GIS Consortium.

Pandazis Jean-Charles, 2002, NextMap: Investigating the future of digital map databases, http://www.ertico.com/download/nextmap_documents/2_164v21.zip.

Raymond A Calss, European Geographical Technologies, 1992, Building Navigable Database for the Real World, In: The 3rd International Conference on Vehicle Navigation & Information System: 585-590.

Thill Jean-Claude, 2000, Geographic Information Systems for Transportation in Perspective, Transportation Research Part C 8: 3-12.

Van Oostrom P., 1995, The Development of an Interactive Multi-scale GIS, INT. J. Geograohic Information Systems, 9(5):489-507.

Worboys, M. F., 1992, Object-Oriented Models of Spatiotemporal Information, Proceedings of GIS/LIS '92 Annual Conference, San Jose, California, USA, American Society of Photogrammetry and Remote Sensing, Bethseda, MA, USA, pp. 825-834.

Worboys, M. F., 1994, A Unified Model for Spatial and Temporal Information, The Computer Journal, 37: 26-33.

Xu A.G., Yang D.K., Cao F.X., et al, 2002, Prototype Design And Implementation for Urban Area In-Car Navigation System, The IEEE 5th International Conference on Intelligent Transportation Systems, 3-6 September, Singapore, 517-521.

Yan Jae-Kwan, Kim Joung-Joon, Hong Dong-Suk, et al, 2005, Development of an Embedded Spatial MMDBMS for Spatial Mobile Devices, Li K. J. and Vangenot C.(eds), W2GIS 2005, LNCS 3833:1-10.

Yonglong Xu, 2000, Development of Transport Telematics in Europe, GeoInfomatica, 4 (2) : 179-200.

Zeiler M., 1999, Modeling Our World, ESRI Press, Redlands, CA.

ACKNOWLEDGMENT

This research is sponsored by the programs :

1.Open Research Fund Program of State Key Laboratory of Information Engineering in Surveying, Mapping and Remote Sensing NO. WKL(07)0305;

2.Province Natural Science Foundation of Hubei NO.2007ABA138