

A NON-PLANAR, LANE-BASED NAVIGABLE DATA MODEL

FOR ITS

Peter Fohl, Kevin M. Curtin, Michael F. Goodchild and Richard L. Church

National Center for Geographic Information and Analysis (NCGIA)
University of California, Santa Barbara
Santa Barbara, USA

Abstract

Digital network databases traditionally employ a fully-intersected planar data model. While this approach has been generally useful and widely accepted, it is only one of many possible paradigms. In an effort to increase the number and improve the quality of network applications, research has been undertaken to develop and employ a fully non-planar representation. Further, the use of lanes as the primary geographic element has provided a previously unattainable level of network detail. These enhancements involve identifying and utilizing new types of feature connectivity. Route finding functions will ultimately be used to provide a test of the utility of the new model, thus a discussion of the challenges posed by routing across lanes is presented. The relative efficiency of planar and non-planar representations is also discussed.

Key words

Non-planar, data modeling, ITS, lanes, networks.

1. Introduction

While the concept of the data model is used in a variety of ways by numerous disciplines, a digital geographic data model is generally defined as an information structure which allows the user to store specific phenomena as distinct representations, and enables the user to manipulate the phenomena when held in the system as data (Faper and Maguire, 1992). Given this definition, the development of such a data model requires a comprehensive knowledge of both the phenomena to be represented and the manipulations which the user intends to apply. Decisions regarding the model components limit the functionality of the database. Each unique feature representation will vary in its ability to serve the needs of any given transportation function. Based on the designers' knowledge of the pertinent transportation functions or processes that will be modeled and the variable functionality of the available data models, the appropriate representation can be chosen.

The primary purpose of this paper is to clearly outline the research and development of a non-planar, lane-based data model for Intelligent Vehicle Highway Systems (ITS). This research is an extension of the planar data modeling documented in the NCGIA Final Report (1994). This paper covers the following topics related to network data modeling:

- A review of planar data models;
- Variability in planar restrictions;
- A comparison of storage requirements for planar and non-planar features;
- The impact of data resolution on the data model design;
- The development of a non-planar, lane based data model;
- The application of network functions to the model.

2. Planar data models

The most commonly used data model for transportation networks is the fully intersected, planar data model. This is an artifact of the prevalence of several widely used, national map databases - in particular, the Topologically Integrated Geographic Encoding and Referencing (TIGER) files from the Bureau of the Census (and their precursors the DIME files), and the Digital Line Graph (DLG) series of products from the United States Geologic Survey (USGS) - and the acceptance of the fully intersected model by a number of industry leading Geographic Information System (GIS) software developers (TIGER/LineO Files 1992). The transportation network has been included in TIGER for a number of reasons. First, the roads are provided to assist enumerators in locating "fugitive" households. The also allows enumerators to correct and maintain the database while they use it. More importantly, the transportation features are often coincident with boundaries of the census geographic regions for which data is collected. Therefore, they must be included to allow the census geographic regions to be well defined. The need to associate census data with these polygonal areas within which individual households are located is satisfied by a fully intersected planar data model. This allows the creation of polygons based on the left and right side attributes of the line segments (including transportation segments). The widespread acceptance and use of the fully intersected model, has been reinforced by ARC/INFO and other software systems which have incorporated this paradigm into their spatial analytic tools (ESRI 1991a).

The presence of a consistent national transportation database is extremely beneficial to the transportation community. However, the limitations that fully-intersected data models place on transportation planning are considerable, and for some applications unacceptable. Of greatest concern is the proliferation of transportation line segments due to the fully intersected data model. At any intersection of linear features - whether physical or statistical - the fully intersected data model demands an intersection with a node marking the end point of each segment. This in turn requires the input and maintenance of the feature attributes for a large number of street segments, which in reality represent a single continuous feature. Often the segments have identical attributes along their lengths, or they have attributes which could be described as an uninterrupted range of values for the single continuous feature. The labor intensive process of entering and maintaining these numerous features has been responsible for introducing a substantial amount of attribute error into such databases. This is especially true when new features or intersections are added. The need to split current features at

that time demands that attributes, such as address ranges, be updated appropriately. This is not an insignificant problem. Moreover, the need to access a large number of segments for any single continuous feature adds to the selection and computation time of transportation functions over the network. The error among the feature attributes is transferred to the network functions applied over the network; these functions can only reasonably be expected to perform as accurately as the attribute data on which they rely. While the obvious positional inaccuracies of several early databases have been a continuous irritant to many users, these errors are primarily a function of the technology used for data capture rather than an artifact of the data model (Spear 1991).

These liabilities have given rise to the current interest in non-planar transportation network models. It is expected that the development of a non-planar prototype will improve on the fully intersected data model in its utility for transportation related applications. While the exploration of this type of data model has been stifled by the inertia of the previously mentioned data collection sources and analytic tools (which have a vested interest in the fully intersected planar model), this also allows for substantial freedom in the decisions made here regarding the development of the prototype.

3. Variability in planar restrictions

Non-planar networks are broadly defined as those networks which allow segments of the network to cross without a network node being located at the intersection. There is, in fact, no implicit or explicit contact between the line segments at the point of intersection. This conceptual model is common in traditional graph theory, and is, in fact, necessary when considering transport modes without well defined, physical routeways. Shipping lanes, for example, often cross in mid-ocean without any node capable of accepting attributes (beyond the approximate location) occurring at the point of intersection. The same is true for airline networks.

Therefore, a non-planar network commonly consists of links and nodes without the mandatory requirement that every intersection be associated with a node. This allows node attributes to be associated with true network intersections but limits the proliferation of unnecessary line segments and unrealistic node placement. Routing of automobiles across such a network is more accurately implemented due to the elimination of impossible turns at multiple-grade crossings (underpass/overpass). Furthermore, users often consider a fully-intersected database to be a less accurate representation of reality, with some nodes solely being artifacts of the data model.

While some unnecessary line segments are eliminated by allowing non-planar intersections, the vast majority of intersections in an urban transportation network are true planar intersections. This suggests that the traditional definition of a non-planar network allows only a relatively small improvement in database size to be achieved, while the functionality of planar systems for polygonal spatial analysis is wholly lost. This research proposes that, if the

planar enforcement is going to be compromised by the data model, the benefit from this relaxed constraint should be maximized. Therefore, the non-planar data model described herein will not utilize nodes to separate line segments at any intersection. This allows each physically continuous, well defined feature to be represented by a single record in the map database. This results in a substantially more compact database with far fewer features. This, in turn, eliminates the repetition of identical attribute data associated with a single feature.

4. Data model feature comparisons

The difference in the number of features between identical networks represented through different data models is a function of the size and connectivity of the network. Thus the reduction in features through conversion from a planar to a non-planar data model will be unique to each network. However, as long as there is a single intersection where two continuous features cross (and if they cross at a point that is not an endpoint for at least one of the features) the non-planar data model will contain fewer features. Thus, if the number of features in a planar network is represented by x_p , then the number of features in a non-planar version of the same network, x_n , will be less than or equal to $x_p - 1$.

This minimum possible improvement in feature quantity is dramatically unimpressive, but the assumption of a network with only one intersection is also totally unrealistic. Commonly transportation networks are well connected, and in urban centers often take the form of a rectangular grid. Because connectivity in rectangular grid networks is consistent within the network, the reduction in features is a function of the grid or matrix size. Below are two statements of the relationship between numbers of features in planar and non-planar data models based on matrix size, and a statement of their equivalence.

4.1 Planar vs. non-planar feature numbers for an $n \times m$ matrix

Consider an $n \times m$ matrix of streets to be a set of n East-West streets and m North-South streets. The total number of non-planar features is $n+m$.

x_p = the number of features in a planar model

x_n = the number of features in a non-planar model

In the planar model each East-West street intersects with each m North-South street creating $m \cdot 1$ arcs. For all East-West streets there are therefore $n(m-1)$ arcs. Similarly there are $m(n-1)$ North-South arcs. Total arcs is thus:

$$n(m-1) + m(n-1) = 2mn - m - n$$

For the case where $n=m$:

$$x_p = 2n^2 - 2n$$

$$x_n = 2n$$

As n increases the number of planar arcs will increase with the square of n , non-planar arcs will grow much more slowly. An algebraic transformation allows one to determine the number of non-planar arcs, given only the number of planar arcs and the special condition that the network is a $n \times n$

matrix, for an arbitrary n :

$$x_n = 1 + \sqrt{1 + 2x_p}$$

With this equation one can find x_n given x_p if we assume that the streets are organized in a $n \times n$ matrix. This is not unrealistic for some urban areas, or for parts of many urban areas. The difference between x_n and x_p is a measure of the improvement in database storage acquired by switching to a non-planar representation.

5. External data model - lanes and spatial information

External data modeling as described by Laurini and Thompson (1992) is the process of selecting real world features for inclusion in the database. As discussed above, the continuous street features will be the smallest element included for display in the prototype data model. However, due to the variation of lane attributes within single street features it is clear that many ITS functions are dependent on the inclusion of individual lanes rather than streets as homogeneous line features. For example, multiple lane streets very often have more than one direction of traffic, lanes vary with respect to turn access, lanes can restrict mode use, and lanes can be made to control speed or other driving parameters. Many secondary attributes which are crucial to ITS applications are based on these physical and mandated attributes of lanes. Among these are travel time by mode, traffic flow, and congestion. Most importantly, the lanes themselves can begin or end at any point along the continuous street feature.

Ideally, lanes would be used as display features as well as primary elements. Unfortunately, several issues of accuracy prohibit lane representation. While GPS technology purports to provide up to centimeter accuracy, these measurements are made in comparison to a benchmark on the Earth's surface. Based on known sources of error, it is generally accepted that locations on the surface of the Earth can only be absolutely known only to within 10 meters. This accuracy is not sufficient to locate a vehicle on a specific lane. Consequently, using lanes as features for display would mislead users regarding the accuracy of the data the system was receiving and dispensing. Therefore, this data model utilizes lanes as the primary element for analysis, and uses streets as the primary display feature.

The available accuracy also affects how spatial information is stored in the internal data structure. All lanes associated with a street or highway follow the same general path, and the variation from that path is not significant relative to GPS accuracy. For these reasons, there is no value in storing lanes as individual polylines. Rather, the street should be stored as a single polyline, with the spatial information for lanes being limited to start points and end points along the relevant street.

The essential fields of the lane table are given in Table 1. Each lane in the database has a unique record in this table, with a corresponding unique identifier stored in the *lane_id* field. The identifier for the street associated with each lane, that is, the polyline the lane follows, is stored in the *street_id* field. Start and end positions of each lane are stored in the *from* and *to* fields, respectively. These positions are given as linear offsets from the beginning of the street polyline.

FIELD	TYPE	DESCRIPTION
<i>lane_id</i>	integer	Unique lane identifier
<i>street_id</i>	integer	Street identifier
<i>from</i>	real	Start position of lane
<i>to</i>	real	End position of lane

Table 1 Lane table.

This method of storing spatial information for lanes is similar to the way linear attributes are modeled using dynamic segmentation in ARC/INFO. Specifically, a lane represented by reference to a particular street polyline with a start and end point is analogous to a linear event referenced to a particular route with a start and end point (ESRI 1991b). However, in ARC/INFO, the route is the basic element and the linear event is an attribute of that element. In the lane based model, the lane is the basic element, while the street polyline is used to store spatial information. For this reason, lanes must contain additional information not associated with linear events: the relationship between lanes. This relationship takes the form of lane connectivity, making the model an integrated representation of the real world, rather than a simple collection of spatial elements.

6. Feature Interconnectivity

A fundamental part of any transportation network data model is the interconnectivity of the basic data elements. If a model does not store, in some fashion, how the elements are connected there can be no analysis of flow between elements. This also precludes routing across the network, a major part of ITS applications.

In the traditional planar data model, with links as the basic data element, interconnectivity is represented by nodes. Links sharing nodes are connected at those nodes (NCGIA 1994). Typically, however, there is a need for further restriction of the interconnectivity. In a model of a street network, planar enforcement can cause a node to exist where there is no actual intersection. An example of this would be a freeway overpass, where a street passes over a freeway on a bridge. Since the street and the freeway each cross over the same point on the ground, planar enforcement requires a node at that location, even though there is no actual intersection. A second common example is the case of legal turn restrictions, where there is an actual

intersection, but drivers are not legally allowed to change directions at that intersection. In both of these instances, the data model must restrict the connection between links.

One standard method used to handle this is the inclusion of a turn restriction table in the data model. This table stores, for each node, the turn restrictions for that node. This turn restriction table can also be used to store turn impedances to more realistically model the real world situation. The ARC/INFO data model uses this method for handling turn restrictions. Each node in the network may have entries in the turn table, identifying an incoming link and an outgoing link, along with information about the impedance of that connection. The maximum number of entries in the table for each node is equal to the square of the number of links sharing that node, but an entry is not required for each link pair. Missing entries are interpreted as having zero impedance, so that it is not necessary to explicitly specify zero impedance for traveling straight through an intersection. Negative impedances are used to disallow connection between two links sharing a node (ESRI 1992).

Another approach is required in the non-planar, lane based data model discussed here. Because intersections are not represented by nodes in the model, turn information must be stored differently. The intention of this model is to accurately represent traffic flow in lanes, so it is appropriate to store turn information as a lane attribute. It is not sufficient, however, to simply list turning options at the end of a lane, because it is frequently possible to turn at more than one point along a lane. In addition, the case of parallel lanes requires the model to allow turns along a linear segment of a lane. In this instance, it is possible to switch lanes at any point along the segment. These segments do not have to continue for the entire length of a lane. Legal or physical barriers may prevent lane changes for certain distances. This is commonly seen in the case of car pool lanes, or when a lane is closed due to an accident or construction (NCGIA 1994). Because of these complexities in the lane based model, the model must be able to store turn information at multiple points and along multiple segments.

The requirement of storing multiple turn possibilities for each lane makes it effectively impossible to include turn information in the same table as lanes. In a relational database this information can be stored in a separate table, similar to the turn table in ARC/INFO. The storage method differs, however, in the method of referencing the turns, because of the complexities discussed above. Two types of turns need to be stored: turn availability at a point on a lane, and linear turn availability along a segment of a lane.

Impedance points must also be introduced into the model. These points indicate places where traffic must slow or stop, even if no turn is made. In the planar data model, this is handled implicitly: even when traveling straight through an intersection a "turn" is made, which may have an impedance value. Again, in the non-planar lane based model, there is no node to use to store impedances, so we must use a different method. Similarities between a turn point and an impedance point strongly suggest that impedance points be

stored with the turns, so we will consider that a third type of turn. Each of these types will be discussed in order.

6.1 Turns at points

While the lane based model is much more complex in terms of connectivity than the planar link-node model, it does have one distinct advantage: lanes only have one direction. In the planar model, movement is possible in both directions along a link, unless it is explicitly prohibited. In the lane based model, movement is only allowed in the fundamental direction of a lane. This implies an extremely straightforward method of storing turn information in a relational database. A table can be used to store a description of each turn possibility in every lane in the database. While this table is likely to be large, proper sorting should make access very efficient.

FIELD NAME	TYPE	DESCRIPTION
lane_id	integer	Turn Origin Lane
turn_id	integer	Unique Turn Identifier
position	real	Start Position on Origin Lane
to_lane	integer	Turn Destination Lane
to_position	real	End Position on Destination Lane
impedance	real	Impedance of the Turn

Table 2 Point turn table.

Consider the basic point turn table definition shown in Table 2. The six fields shown are the essential elements of a turn table in this model. The first field, *lane_id*, contains the identifier of the lane. This value references a unique entry in the lane table (Table 1). The second field, *turn_id*, is the identifier of the turn, unique within each lane. The field pair consisting of *lane_id* and *turn_id* is unique in the turn table, and can be used to reference turns globally in the database. The positional information of the turn along the lane is stored in the *position* item, giving the offset along the street containing the lane at which the turn is located.

The destination point of the turn is given in the next two items. *To_lane* indicates the destination lane, and references the *lane_id* item in the lane table. The *to_position* field specifies the location along the *to_lane* where the connection occurs. The final item in the table, *impedance*, contains the impedance value of making that turn. Additional fields can be added to this table containing time-based restrictions or impedances, or other turn attributes. If the table is sorted with *lane_id* as the primary key and *position* as the secondary key, the first turn along a lane can be found using a binary search algorithm, then turns can be processed in the order in which they actually occur in the physical network. Disallowing turns in the database can be handled quickly by "turning off" the turn by assigning a negative value to the *impedance* field.

6.2 Turns along segments

Turns along segments can be represented in a fashion similar to point turns, with a table containing one entry per turn. However, differences in the turn types require a few changes in the fields. Rather than a simple turn position, the linear turns require a start position and an end position. In addition, the turn destination depends on where along the segment the turn is made, rather than being a single position on the destination. The turn also not only has an associated impedance, as with point turns, but also a travel distance required to complete the turn. A basic linear turn table definition is shown in Table 3.

FIELD	TYPE	DESCRIPTION
lane_id	integer	Lane on which the turn occurs
turn_id	integer	Unique turn identifier
start	real	Location of the beginning of the turn on the lane
end	real	Location of the end of the turn on the lane
to_lane	integer	Destination lane of the turn
to_offset	real	Location on the <i>to_lane</i> corresponding to the start position of the turn
distance	real	Linear distance required to complete the turn
impedance	real	Impedance of the turn

Table 3 Linear turn table.

Four of the items in this table are identical to items in the point turn table previously discussed (Table 2): *lane_id*, *turn_id*, *to_lane* and *impedance*. These items have the same meaning in both tables. The *position* and *to_position* fields in the point turn table are missing in this table, with the analogous information stored in other fields. *Start* and *end* together represent the location of the turn in the lane, with *start* giving the beginning location of turn availability and *end* giving the ending location. This indicates that the turn may be taken at any point along the lane between *start* and *end*.

In the point turn table, the destination point is given as a single location (*position*) along the *to_lane*. The situation is more complex in the linear turn table. First, the destination point is dependent on the location where the turn is actually taken, not just where the turn availability begins. Further, some distance must be traveled along the lane before the turn can be completed. The *to_offset* and *distance* fields provide the information necessary to calculate the destination point from a turn location between *start* and *end*. *To_offset* gives the location along *to_lane* that is longitudinally equivalent to *start*. In other words, *to_offset* would represent the destination point of a turn taken at *start* if the distance requirement of the turn was zero. Distance represents the distance requirement of the turn. With this information, the final destination point along *to_lane* can be calculated using the following formula:

$$\text{Location} = \text{to_offset} + (\text{turn location}) \cdot \text{start} + \text{distance}.$$

As with the point turn table, proper sorting of the linear turn table must be maintained for efficient access. Clearly the primary key must again be *lane_id*, so that the set of turns for a lane can be quickly found. Further sorting criteria are less obvious. A common need in an application using this data model would be to find the set of turns available at a given point along the lane. If the table is sorted with start as the secondary key, all turns with a start value higher than the current position can be quickly eliminated. However, since linear turns can, and often will, extend the entire length of a lane, each turn will start less than or equal to the current position would need to be examined in turn.

6.3 Impedance points

The previous discussions ignore a fundamental element of transportation networks, impedance associated at a point where no turn is made. This impedance could represent a stop sign, slowdown due to merging traffic, or other legal or physical restriction. This type of impedance is handled in this data model by the use of impedance points. Because impedance points share two of the key attributes of point turns, namely a single location and an associated impedance, it is a simple matter to store them in the point turn table. The two attributes they do not share, *to_lane* and *to_position*, can be ignored or made equal to *lane_id* and *position*, respectively. Either method is sufficient to distinguish an impedance point from a turn.

The fundamental difference between impedance points and point turns is the way that they are applied. A point turn represents an option available at that location on the lane, generally with an associated impedance, while an impedance point is automatically applied when a particular location is reached, if no turn option is taken at that point. Impedance points can also be used to represent the end points of lanes, as a point of infinite impedance. While lane endings are stored in the lane table, it may prove desirable to include this information in the turn tables as well, for some applications.

7. ITS Functions

Any research into the development of an effective digital map database for transportation purposes must accept the responsibility to provide for the integration of ITS methods and functionality into the database. It is, in fact, the inability of current data models to efficiently support this flexibility that has initiated the research at hand. According to NCGIA (1994) it is well recognized that "most of the anticipated functions of ITS rely on the existence of an accurate, dependable map database, with sufficient information on network connectivity and driving conditions to allow solution of such problems as determination of the shortest path between a given origin and destination." The map database and ITS functions will be integrated, and the ways in which to do so most effectively must be considered during the development process.

Transportation planning is, however, a broad based discipline with many specializations and countless applications. Each of these deserves a voice in determining which functions will compose a fully functional ITS, and what needs must be satisfied by the underlying digital map database. In an effort to record these diverse needs the U.S. Department of Transportation (USDOT) has produced a Program Plan for the Intelligent Vehicle Highway Systems Program. As discussed in the NCGIA Final Report, the individual systems given as potential ITS applications vary in the extent to which they will interact with the digital map database. Some will be entirely independent of the digital map database, but will rely on the communications and information transfer infrastructure that will necessarily accompany a fully functional ITS. For example, several types of collision avoidance systems and in-vehicle safety devices do not interact with the navigable map database. At most these systems will receive and transmit location information about vehicles. Trip navigation is not a concern. Other functions will be associated more closely with specific static positions on the map database but will still be relatively unconcerned with the ability to navigate across it. These include secondary information sources regarding traveler services, safety checkpoints, electronic payment stations, and various administrative functions. Those functions that do interface with the navigable map database are primarily associated with vehicle routing.

7.1 Vehicle Routing

The ITS must be able to provide an initial trip route (based on destination information provided by the user along with historical travel time data and dynamic incident information), update that route as conditions change, guide the user across the route, and inform the user of any pertinent information. Normally, shortest path algorithms developed in mathematical graph theory are used in routing applications on planar networks. Traditional planar networks correspond very well to mathematical graphs, so the use of graph theory algorithms is straightforward. A mathematical graph is an object consisting of a set of elements called vertices and a set of pairs of vertices called edges (McHugh 1990). Relating this to a planar transportation network, vertices correspond to nodes and edges correspond to links. With the non-planar model presented here, existing algorithms cannot be so directly applied.

Graph theory shortest path algorithms such as Dijkstra's (Dijkstra 1959) rely on the structure of a graph, with each edge connected to other edges only at its endpoints. This is not the case with the lane based model, where turns are possible at many points along a lane. Further, as discussed in the previous sections, lanes are frequently connected by linear lane change segments, rather than simple intersections. Because of this, routing on the lane based model requires either the development of a new shortest path algorithm or a method of fitting a mathematical graph to the lane based model so that existing algorithms can be used. The most practical approach appears to be the latter.

The chief difficulty in fitting a mathematical graph to the lane based model is dealing with continuous lane change segments. Because a turn can occur at any point along a segment, explicit representation of the segment in a graph would require an infinite number of vertices, clearly an impossibility. One solution would be to define a minimum edge length, with a continuous segment represented by a series of these edges so that a change can be made to adjacent lanes at any vertex along the series. For example, a 1 mile segment could be represented by 528 10 foot edges. To avoid artificially imposing turn restrictions, this edge length would need to be significantly less than the distance required to change lanes. For a typical highway, each lane would need to be represented, for routing purposes, by a large number of edges. The time required to complete Dijkstra's algorithm, for instance, is $O(V^2)$, where V is the number of vertices (McHugh 1990). It is easy to see that this would quickly become too large for real time routing.

An alternative to this would be to assume that a lane change is made as soon as possible if it is made at all. Using this assumption, a vertex would be placed at the first possible position along the lane, with that vertex being the only connection between the lanes. In effect, this dynamically reduces the linear turn possibility to a turn at a point. It should be noted that this assumption could be problematic in situations with large numbers of lane blockages. For instance, on a two lane road, if each lane is blocked at several points the road would be impassible without multiple lane changes. This could be avoided by allowing lane changes at points where lane conditions (i.e. congestion, construction, accidents) change drastically.

8. Conclusions

This paper represents a significant departure from the traditional planar, centerline based, transportation network. A fully non-planar representation is adopted here in order to more realistically portray human perceptions of real world transportation networks. It is clear that non-planar networks allow a substantial reduction in the number of features to be maintained in the database, and eliminates redundancies among feature attributes. The extent to which this improvement is realized is a function of the size and connectivity of the network. The inclusion of lanes as the primary geographic element allows a higher level of detail than previously achieved, and permits additional road condition information to be utilized, such as the closure of specific lanes due to accidents or construction.

At this time a prototype non-planar network has been developed and is being used to reference lane features for a test area. The test area includes a number of urban and suburban road configurations which represent local, arterial, and highway traffic conditions. Further research will include the integration of GPS spatial referencing, simulated real-time traffic conditions, and in-vehicle navigation tools to provide a dynamic routing system.

Acknowledgments

This work was completed under the CALTRANS/NCGIA Memorandum of Understanding which initiated a research project to develop the concepts, methods, and techniques of a navigable map database for ITS.

References

- Dijkstra, E. W. (1959) A Note on Two Problems in Connexion with Graphs. *Numerische Mathematik*. Berlin. Vol. 1, No. 5, pp. 269-271.
- ESRI (1991a) ARC/INFO Data Model, Concepts, & Key Terms. ARC/INFO User's Guide. Redlands, CA.
- ESRI (1991b) Dynamic Segmentation. ARC/INFO User's Guide. Redlands, CA.
- ESRI (1992) Network Analysis. ARC/INFO User's Guide. Redlands, CA.
- Freundschuh, S. M., M. D. Gould, D. M. Mark (1989) Issues in Vehicle Navigation and Information Systems. National Center for Geographical Information and Analysis, State University of New York at Buffalo. Technical Paper 89-15.
- Laurini, R., D. Thompson (1992) Fundamentals of Spatial Systems. Academic Press, San Diego, CA.
- McHugh, J. A. (1990) Algorithmic Graph Theory. Prentice Hall, Englewood Cliffs, New Jersey.
- NCGIA (1994) Final Report - CALTRANS Agreement 65T155. National Center for Geographic Information and Analysis. Santa Barbara, CA.
- Raper, J. F., D. J. Maguire (1992) Design Models and Functionality in GIS. *Computers & Geosciences*. Vol. 18, No. 4, pp. 387-394.
- Spear, B. D. (1991) Issues and Recommendations From the Standpoint of Transportation Analysis. Research and Special Programs Administration, U.S. Department of Transportation. TIGER Enhancement Technical Working Group, Meeting Minutes, October 27, 1991. GIS/LIS '91 - Atlanta, GA.
- TIGER/Line™ Files (1992) Geographic Objects in the TIGER/Line (TM) Files. U.S. Bureau of the Census, Washington, DC.

Contact address

Peter Fohl, Kevin M. Curtin, Michael F. Goodchild and Richard L. Church
National Center for Geographic Information and Analysis (NCGIA)
University of California, Santa Barbara
3510 Phelps Hall
Santa Barbara, CA 93106-4060
USA
Phone: +1-805-993 8652
Fax: +1-805-993 8617
E-mail: foht@ncgia.ucsb.edu
E-mail: curtin@ncgia.ucsb.edu
E-mail: good@geog.ucsb.edu
E-mail: church@geog.ucsb.edu