

Spatial Network Modeling for Databases

Virupaksha Kanjilal & Markus Schneider*

University of Florida

Department of Computer & Information Science & Engineering

Gainesville, Florida, USA

{vk4, mschneid}@cise.ufl.edu

ABSTRACT

Spatial networks like transportation, power, and pipeline networks are a ubiquitous spatial concept in everyday life and play an important role for navigational and routing purposes. Database support for large spatial networks in order to represent, store, query, and manipulate them is rare. Our paper aims to provide the beginning of a conceptual, abstract, and formal model of spatial networks, called Spatial Network Algebra (*SNA*), that includes types and operations and is supposed to serve as a specification for their later implementation in spatial database systems and GIS. Finally, we show how our spatial network concepts can be embedded into an SQL-like query language.

1. INTRODUCTION

Spatial networks like road networks connecting cities, rail networks connecting railway stations, pipeline networks carrying water to our houses are ubiquitous in everyday life. Spatial networks are *spatially embedded graphs* created by the interconnection of spatial elements such as spatial lines and spatial points. Apart from their use in spatially oriented disciplines like geography, cartography, Geographical Information Systems (GIS), and spatial databases systems, spatial networks also find application in transport and navigation as in GPS devices or in traffic forecasting models.

Database support is necessary to efficiently utilize the large volumes of data in any spatial network. But till now, database support for spatial networks is rare and neglected. Available GIS implementations of spatial networks only use the database system to deliver the basic components of the network so that an in-memory network can be built in a middleware layer. This approach fails to take advantage of the benefits provided by the database system like query processing, concurrency control, and transaction management.

To integrate spatial networks in a database context as a first class citizen, we need a formal model of spatial networks

*This work was partially supported by the National Science Foundation under grant number NSF-IIS-0812194.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

SAC'11 March 21-25, 2011, TaiChung, Taiwan.

Copyright 2011 ACM 978-1-4503-0113-8/11/03 ...\$10.00.

which will serve as the specification for any implementation. The goal of this paper is to create a data model for networks embedded in space with its own type definitions. Due to space constraints, we will only introduce a few of the basic operations and predicates on them. Finally we show a few queries in an SQL-like query language that take advantage of the operations and predicates on a spatial network. This model is a part of a complete algebra for spatially embedded graphs called the *Spatial Network Algebra* (SNA).

In our approach, points in the Euclidean plane which are a part of the network have thematic properties associated to them. The thematic properties distinguish components of a network, i.e., all the points with equal thematic properties belong to the same network component. Interior and exterior points of a network can be distinguished based on the thematic properties. In particular, this approach enables us to consider attributes of single points (space based view) but also provides access to collections of points having equal attributes (object based view).

Section 2 gives an overview of the various approaches to modeling spatial network. Section 3 introduces the Spatial Network Algebra as our model of spatial networks. Section 4 defines a few important operations and predicates on our spatial network model. In Section 5 we introduce an SQL-like query language that can be used to create or query spatial networks.

2. RELATED WORK

In this section, we confine ourselves exclusively to modeling approaches to spatial networks. A reasonable and popular concept is to model spatial networks as planar graphs in order to capture their structure and connectivity. Queries like the *shortest path* or the *maximum flow* can be directly mapped to well known graph problems for which algorithms exist. For example, the shortest path problem may be solved by Dijkstra's algorithm, and the maximum flow problem can be solved by the Ford Fulkerson algorithm. This approach to modeling networks has been taken in [4] and [6]. The model in [4] has been designed for embedding graphs in databases and not specifically for spatial graphs. The work in [6] models road networks by defining road components based on their properties and the potential actions they may perform. The authors term this as an "affordance-based" theory of networks. Though graph modeling has some advantages, real world networks are too complex to be represented by simple graphs [5].

Routes are essential concepts in a spatial network, as the models in [3, 1] show. Routes correspond to paths in a graph

and to roads in a transportation network. In these models, a network can be explicitly stored in a database which enables the formulation of powerful queries and efficient execution. However these approaches cannot store and represent the thematic attributes of the network components. The smallest network unit in these cases is a road or a junction. In our model, we assume every point of the network is marked with its thematic attributes, and we aggregate all points with equal attributes to form network components.

3. THE SPATIAL NETWORK MODEL

In this section we introduce our model of spatial networks. We begin by providing an intuitive description of spatial networks in Section 3.1. In Section 3.2 we give the formal definitions of network components and the network itself.

3.1 What are Spatial Networks?

Spatial networks are a ubiquitous spatial concept. We use transportation networks like road networks for cars, buses, and taxis or railway networks for trains and metro every day. Water pipelines and power networks supply our houses with valuable resources. If we abstract from these particular kinds of networks, we can say that the primary purpose of spatial networks is to provide a spatially constrained environment for materials (in the broadest sense) to move or flow through them.

From a modeling perspective, a spatial network can be seen as a *spatially embedded and labeled* graph. This means that a spatial network has geometric and thematic aspects. A spatially embedded graph is a graph whose vertices are mapped to points in space and whose edges are mapped to arcs which have end points that are images of two vertices [2]. Geometric aspects comprise junctions, channels, boundary points, and crossover points that are embedded in space and characterized by precise locations. Thematic aspects specify the semantics (like names, functions, or properties) of geometric components and are attached to them in terms of annotations or *labels*.

Figure 1a shows a map depicting a network of roads. Figure 1b is our simplified representation of the same road network. Some important roads from Figure 1a have been abstracted as curves which form the edges of the graph. There are various points of interest on the network which are analogous to nodes in a graph. We term these points as *interaction points*; they have been marked by circles in Figure 1b. Interaction points include junction points, crossover points, and boundary points. Junction points are network components which mark locations where two (or more) roads intersect. The point j_1 in Figure 1b is a junction because at this point the roads *W Newberry Road* (shown as l_1 in Figure 1b) and *SW 75th street* (shown as l_6 in Figure 1b) intersect. Cars can switch from one road to another only at junction points. This is opposed to crossover points where two roads interact but one road forms a bridge over the other. For example, points c_1 , c_2 , and c_3 in Figure 1b are crossover points because at these points, the *Interstate 75* (shown as l_5 in Figure 1b) forms a bridge over *W Newberry Road*, *SW 20 Avenue*, and *SW Archer Road* (represented as l_1 , l_3 and l_4 respectively in Figure 1b). Cars cannot switch directly from one road to another at crossover points as the interacting roads are separated by a vertical gap. Junction points and crossover points are not mutually exclusive; both may exist at the same point. As an example, three roads may interact at the

same point, but only two of them actually intersect forming a junction, while the third one may form a bridge over it. Junction points and crossover points are identified in a 2D representation of a spatial graph by the degree of their nodes. They both have a node degree more than one. But it is not possible to distinguish between a junction point and a crossover point from the spatial graph alone due to the flat, two-dimensional network representation. Boundary points are another kind of interaction points whose node degree is equal to one. Points b_1, \dots, b_{11} are boundary points as they form the limiting points of the network.

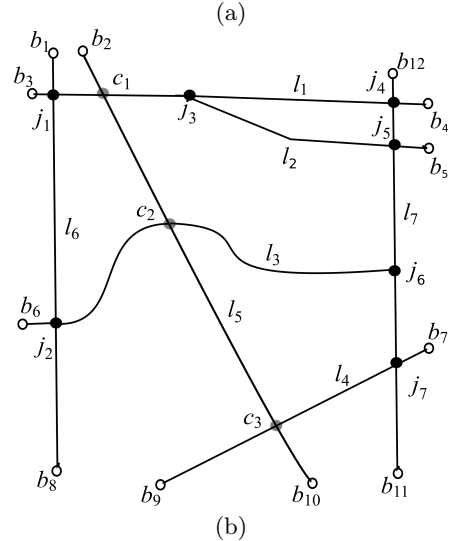


Figure 1: An Example of a Spatial Network

The interaction points in a spatial network are connected to each other by what we term as *channels*, and they are the passages through which material is transported. A finite sequence of connected edges in a spatially embedded graph forms a channel. In Figure 1a, the roads are channels. The pipes in a pipeline network and the tracks in a railway network are examples of channels. A channel has exactly two end points. The end points of a channel may intersect with other channels or remain free. As an example, one of the end points of channel l_3 intersects with channel l_7 at point j_6 . On the other hand, if an end point does not intersect with any other channel, it represents a dead end. A dead end may be considered as an extremity of a network, and we call it *boundary* of the network. All the points marked by hollow circles in Figure 1b are network boundary points.

A channel is not only specified by the geometry of the edges; in addition, it is defined by a name attached to its geometry. For example, highway I75 cannot be identified in a network by its geometry alone. To identify it, we have to locate all the points in the Euclidean plane which have the name “I75”. In our model of spatial networks, we assign thematic attribute values to all points of a channel. The attribute values can be used to identify network components like channels and interaction points. We call these values *labels*, and each point in the Euclidean plane is mapped to a set of labels. Any label is a tuple of the form $(id_attr, theme_attr)$. The *id_attr* part of a label contains values which uniquely identify a particular channel in a network. This can be a channel name (e.g. I75) or a channel identifier (e.g. pipe no. 6). All points in the Euclidean plane having the same *id_attr* part of the label are part of the same channel. The *id_attr* part is always at the first position of a label. The *theme_attr* part of the label models the thematic attributes of a point on the plane. The *theme_attr* values may have a simple type, such as integers, or a string, or a complex type, representing a combination of n values, for example. Examples of thematic attributes range from speed limits of a road to the capacity of an oil pipe.

We assume that each label type A contains an element \perp_A (called undefined or unknown). It is called the *exterior label*, and the outside area of a network is labeled by \perp_A . For the Cartesian product of two types A and B we let $\perp_{A \times B} = (\perp_A, \perp_B)$, and for the union of A and B we identify \perp_A, \perp_B , and $\perp_{A \cup B}$ (that is, we take the coalesced sum). If no ambiguities can arise, we sometimes omit the type index and simply use \perp . In all network visualization tools, coloring and markings differentiate channels. This is similar to our assignment of thematic attributes to points in a spatial network.

3.2 Definition of Spatial Networks

As we have motivated in the previous subsection, a spatial network is a spatially embedded and labeled structure. We assume that semantically relevant thematic labels are attached to each point in the Euclidean plane. We call this many-to-one mapping between spatial points and labels *spatial mapping*. A correct assignment of labels to points in the Euclidean plane helps us identify channels, junctions, and crossovers, and also distinguish coexisting junctions and crossovers. Junction points and crossover points are formed by the interaction of two or more channels at a point in the plane; thus, these points are labeled by the combination of the labels of the interacting channels. Junction points express connectivity of the network. This means a car standing at a junction of a road is allowed to go to any of the roads connected to that particular junction. In order to maintain the connectivity information, a set of channel labels for a point indicates the intersection of the channels belonging to the labels in the set. Figure 2 shows the road network from Figure 1 with an appropriate labeling. The road network shown has seven roads represented by the channels in Figure 1 with labels from the range $l_1 \dots, l_7$. Thus, in this case, the road network is a spatial mapping of type A , where $A = \{l_1, l_2, l_3, l_4, l_5, l_6, l_7\}$. The junction point j_1 in Figure 1 is formed by the intersection of the channels l_1 and l_6 and is represented by the set $\{l_1, l_6\}$. This notation expresses that both channels l_1 and l_6 are present at the point j_1 . If such a set contains only one label, the corresponding point belongs

only to a single channel without a junction involved.

A crossover point means that two or more channels interact but they do not geometrically join. This is modeled as a set of disjoint sets of labels. For example, crossover point c_1 , as shown in Figure 1, is formed by the interaction of channels l_1 and l_5 . At the point c_1 , the interacting channels are represented as singleton sets $\{l_1\}$ and $\{l_5\}$. Hence, the label for the crossover at point c_1 is modeled as the set of these singleton sets, namely $\{\{l_1\}, \{l_5\}\}$ (Figure 2).

Sometimes, junctions and crossovers coexist. This means that some channels can form a bridge over a junction point (like a highway bridge passing an interstate) resulting in the creation of a crossover along with the junction at the same Euclidean point. It may also be the case that two or more junctions (formed by disconnected sets of channels) exist at the same point but are separated from each other by a vertical gap. In these two situations, a junction point and a crossover point appear together, and we call such points *dual interaction points*. It is important to distinguish between junction points, crossover points, and dual interaction points in a network because they have an impact on the computation of shortest paths and other network queries. As an example of a dual interaction point, consider a spatial mapping of type A with $A = \{l_{A,1}, l_{A,2}, l_{A,3}, l_{A,4}\}$ and suppose that the channels $l_{A,1}$ and $l_{A,2}$ form a junction, which is represented as $\{l_{A,1}, l_{A,2}\}$, and that the channels $l_{A,3}$ and $l_{A,4}$ form another junction, which is represented by $\{l_{A,3}, l_{A,4}\}$, at the same point. The two junctions form a crossover with each other and thus represent a dual interaction point. This dual interaction point is represented by a set of sets of labels, namely $\{\{l_{A,1}, l_{A,2}\}, \{l_{A,3}, l_{A,4}\}\}$.

In summary, we see that a point of a channel that is neither a junction point nor a crossover point can be modeled as an element of a label set A ; an example is $l_{A,3}$. The modeling of a junction requires a set of labels indicating the participating channels, i.e., an element of 2^A , e.g., $\{l_{A,1}, l_{A,2}\}$. The modeling of a crossover point requires a set of sets of labels indicating the participating channels, i.e., an element of 2^{2^A} , e.g., $\{\{l_{A,1}\}, \{l_{A,2}\}\}$. The modeling of different junctions that form a crossover at the same location requires a set of sets of labels indicating the participating junctions, i.e., an element of 2^{2^A} , e.g., $\{\{l_{A,1}, l_{A,3}\}, \{l_{A,2}, l_{A,4}\}\}$. We see that points of the Euclidean plane can be either mapped to A , 2^A , or 2^{2^A} . In order to obtain a unique mapping for all points of the Euclidean plane, we take the most general case and make a spatial mapping a function which maps points from R^2 to 2^{2^A} . Hence, we obtain the sets $\{\{l_{A,3}\}\}$, $\{\{l_{A,1}, l_{A,2}\}\}$, $\{\{l_{A,1}\}, \{l_{A,2}\}\}$, and $\{\{l_{A,1}, l_{A,3}\}, \{l_{A,2}, l_{A,4}\}\}$ in the examples above. The set $\{\{\perp\}\}$ characterizes points having no label and makes the spatial mapping function a total function. Definition 1 provides the specification of a spatial mapping.

Definition 1 Let A be a label type. A spatial mapping of type A is a total mapping $\nu : \mathbb{R}^2 \rightarrow 2^{2^A}$.

The set of all spatial mappings of type A is denoted by $[A]$, that is, $[A] = \mathbb{R}^2 \rightarrow 2^{2^A}$. When applied to a set X , the function ν is iteratively applied to all the elements of X , that is, $\nu(X) = \{\nu(p) | p \in X\}$. The concept of spatial mappings is too general for spatial networks. In other words, not every spatial mapping represents a spatial network. We

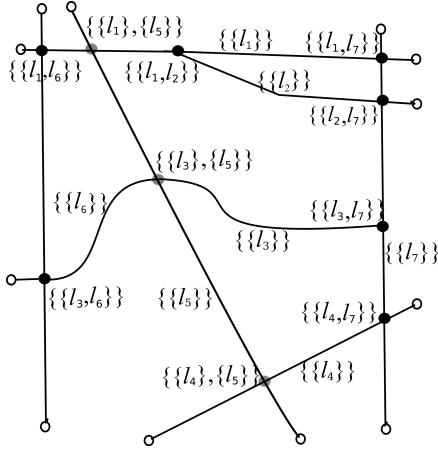


Figure 2: A Spatial Network $[A]$ with labels

have to impose certain restrictions on spatial mappings as described in the following. The idea is to infer and distinguish channels, junctions, and crossovers from the label information. Points which have a label other than $\{\perp\}$ belong to a channel. A crossover point is indicated by a set containing at least two sets of channel labels. To identify junction points, we look into each set of labels. If any one of the sets has more than one channel label, it means that the channels which belong to those labels form a junction. Definition 2 identifies channels, junction points, and crossover points in a spatial mapping.

Definition 2 Let ν be a spatial mapping of type A . Then

- (i) $L(\nu) = \{p | p \in \mathbb{R}^2 \wedge \nu(p) \neq \{\perp\}\}$ (channels)
- (ii) $J(\nu) = \{p | p \in \mathbb{R}^2 \wedge (\exists l \in \nu(p) : |l| > 1)\}$ (junctions)
- (iii) $C(\nu) = \{p | p \in \mathbb{R}^2 \wedge |\nu(p)| > 1\}$ (crossover)

$L(\nu)$ contains all the points in the Euclidean plane which are part of a channel. Individual channels can be identified by the label information, as the id_attr part of the label uniquely identifies a channel. All the points of the same channel have the same id_attr part in their label. Thus, each channel may be distinguished by grouping the points in $L(\nu)$ by the id_attr part of the label. This necessitates a look into the labels and an extraction of parts from them. We can assume that an element of the label type A is a sequence of label attribute values and that each such value is of a (possibly different) set A_i . That is, $A = \times_{i=1}^k A_i = A_1 \times \dots \times A_k$. Let $I = \{1, \dots, k\}$, $S = \{j_1, \dots, j_n\}$, and $S \subseteq I$. To extract selected attribute values from a label, we define a projection operator Π as follows:

$$\Pi_S : \times_{i \in I} A_i \rightarrow \times_{j \in S} A_j$$

with $\Pi_S(a_1, \dots, a_k) = (a_{j_1}, \dots, a_{j_n})$.

Next we define a function called Id_Attr to extract all the id_attr label parts actually present in a spatial mapping. It takes as argument a spatial mapping ν of type A and computes the set of all id_attr values by using the projection operator Π . As the id_attr attribute is assumed to be always the first attribute a_1 in a label, we use $\Pi_{\{1\}}$ to extract its value. Further, we generalize function applications from

elements to sets of elements. Let $f : X \rightarrow Y$ be a function, and let $B \subseteq X$. Then we allow to use the notation $f(B)$ which is given as $f(B) = \{f(x) | x \in B\}$. This is here applied to a spatial mapping ν . The function Id_Attr is now defined for $\nu \in [A]$ as:

$$Id_Attr(\nu) = \{\Pi_{\{1\}}(l) | s \in \nu(L(\nu)), e \in s, l \in e\}$$

Each channel has a unique id_attr value; thus, all points which belong to the same channel have the same id_attr value in their labels. Definition 3 identifies all channels in a spatial mapping by defining the two functions $Channel$ and $Channels$. The function $Channel$ gets an id_attr value as input and determines all points in the Euclidean plane that have this value in their labels and thus form a particular channel. The function $Channels$ gets a spatial mapping as input and collects all channels by iterating over all of its id_attr values. Points representing junction points or crossover points are part of interacting channels; hence, they appear in more than one channel.

Definition 3 Let ν be a spatial mapping of type A , and let $l \in Id_Attr(\nu)$. Then

- (i) $Channel(l) = \{p | p \in L(\nu) \wedge \exists s \in \nu(p) \exists e \in s : \Pi_{\{1\}}(e) = l\}$
- (ii) $Channels(\nu) = \{Channel(l) | l \in Id_Attr(\nu)\}$

For the definition of a spatial network, we have to consider its underlying line-shaped geometric structures. This requires the concept of a *simple line* which we give in Definition 4.

Definition 4 The set *sline* of all *simple lines* in the Euclidean plane is defined as:

- $$sline = \{L \subset \mathbb{R}^2 |$$
- (i) $\exists f : [0, 1] \rightarrow \mathbb{R}^2 : L = f([0, 1])$
 - (ii) f is a continuous mapping
 - (iii) $|f([0, 1])| > 1$
 - (iv) $\forall a, b \in]0, 1[, a \neq b : f(a) \neq f(b)$
 - (v) $\forall a \in \{0, 1\} \forall b \in]0, 1[: f(a) \neq f(b)$

Conditions (i) and (ii) require the existence of a continuous function that generates the simple line. Condition (iii) avoids the anomaly that all elements of the unit interval are mapped to the same point. Condition (iv) states that a simple line is not allowed to be self-intersecting. Condition (v) requires that a simple line is not self-touching.

We are now able to define a *spatial network* of type A as a special spatial mapping of type A (Definition 5).

Definition 5 A *spatial network* of type A is a spatial mapping ν of type A such that

- (i) $\forall L \in Channels(\nu) : L \in sline$
- (ii) $\forall j \in J(\nu) : j \in L(\nu)$
- (iii) $\forall c \in C(\nu) : c \in L(\nu)$
- (iv) $\forall c \in C(\nu) \forall s_1, s_2 \in \nu(c) \forall l_1 \in s_1 \forall l_2 \in s_2 : \Pi_{\{1\}}(l_1) \neq \Pi_{\{1\}}(l_2)$

Condition (iv) states that, in case of a crossover, the participating channels and/or junctions must be different since they cannot be physically present at more than one junction. This means that labels like $\{\{l_5\}, \{l_5\}\}$ or $\{\{l_1, l_5\}, \{l_3, l_5\}\}$

are invalid. The condition checks whether the *id_attr* values of the channels and/or junctions at each crossover point are disjoint.

We do not specify constraining topological relationships between different channels of a spatial network. This means that different channels may meet, partially overlap, or one channel may be contained in another channel. For example, in a road network, many roads carry several names. This is, for instance, the case for U.S. Route 441 which is a spur route of U.S. Route 41.

A channel L with a describing function $f_L : [0, 1] \rightarrow \mathbb{R}^2$ has two *end points* $f_L(0)$ and $f_L(1)$. Dual interaction points, represented by $D(\nu)$, indicate the co-existence of junctions and crossovers. That is, $D(\nu) = J(\nu) \cap C(\nu)$. If $D(\nu) = \emptyset$ holds, the spatial network does not have dual interaction points. If additionally $C(\nu) \neq \emptyset$ holds, this means that crossover points are passed by single channels.

The boundary points of a spatial network ν are those end points of the channels that are not shared by other channels. Let $Channels(\nu) = \{L_1, \dots, L_n\}$ that are described by functions f_{L_1}, \dots, f_{L_n} . Let $E(\nu) = \bigcup_{i=1}^n \{f_{L_i}(0), f_{L_i}(1)\}$ be the set of end points of all channels of ν . The set $S(\nu) \subset E(\nu)$ of those points that are shared by more than one channel is given as

$$S(\nu) = \{p \in E(\nu) \mid \text{card}(\{f_{L_i} \mid 1 \leq i \leq n \wedge f_{L_i}(0) = p\}) + \text{card}(\{f_{L_i} \mid 1 \leq i \leq n \wedge f_{L_i}(1) = p\}) \neq 1\}$$

Then the boundary points $B(\nu)$ are given as $B(\nu) = E(\nu) - S(\nu)$.

4. OPERATIONS ON SPATIAL NETWORKS

A large number of interesting operations on spatial networks can be designed which assist in posing queries on spatial networks. Here we only describe a few of them including *Routes* (Section 4.1), *Length* (Section 4.2), and *ShortestRoute* (Section 4.3), which are classical network operations.

4.1 Routes between Two Network Points

A *route* is a course (way, path, connection) one can take in order to reach a second location from a first location. Given a spatial network, a route connects two points of the network through an alternating sequence of channels and junctions of the same network. A route between the two cities Atlanta and Gainesville is an example. Finding routes is an important feature of spatial network applications as the locations of moving objects in a network are stored with respect to a particular route. There can be possibly a large number of routes between two points in a network. We consider routes to be spatial networks with certain constraints. As a route connects two points p and q , the route starts at p and ends at q ; i.e., p and q are the boundary points of the spatial network which represents a route. To prevent any discontinuity in the route, it can only have exactly two boundary points.

In our model, we define an operator called *Routes* which takes two points inside a spatial network and creates the set of all possible connections between them. The signature of the *Routes* operator is $Routes : [A] \times \mathbb{R}^2 \times \mathbb{R}^2 \rightarrow 2^{[A]}$. This operator returns the set of all spatial subnetworks representing routes between two selected points over a given spatial network. All points of each returned spatial network, i.e., route, are a subset of the points of the original

network. Figure 3 shows the resulting routes when the operation $Routes(\nu, j_2, j_6)$ is performed on the network depicted in Figure 2 to calculate the paths from j_2 to j_6 . Given a spatial network $\nu : [A]$ and two points $p, q \in L(\nu)$, we define the *Routes* operator as

$$Routes(\nu, p, q) = \{\nu' \mid \begin{array}{l} \text{(i)} \quad \nu' : [A] \text{ is a spatial network} \\ \text{(ii)} \quad L(\nu') \subseteq L(\nu) \\ \text{(iii)} \quad \forall l \in \nu'(J(\nu')) : |l| = 2 \\ \text{(iv)} \quad p, q \in B(\nu') \\ \text{(v)} \quad |B(\nu')| = 2 \end{array}\}$$

Condition (i) states that every route is a spatial network. Condition (ii) ensures that every route ν' is a subnetwork of ν . Condition (iii) requires that each junction of a route must have a degree of exactly two. Condition (iv) states that p and q are boundary points of the route ν' . Condition (v) ensures that p and q are the *only* boundary points of ν' .

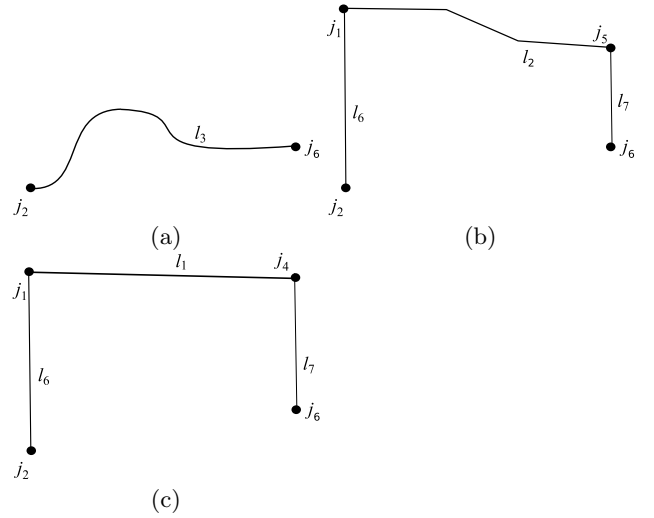


Figure 3: The routes returned by executing $Routes(\nu, j_2, j_6)$ on the spatial network shown in Figure 1b

4.2 Length of a Channel

Length is an important operator which calculates the length of a channel. It yields a real value as a result. The *Length* operator has the signature $[A] \rightarrow \mathbb{R}$. But only channels are allowed as input parameters. A channel must be integrable and bounded. This is always the case as the definition of channels requires that a channel is continuous and the describing function is bounded in the interval $[0, 1]$. To calculate the length, we divide the entire channel into infinitesimally small chord approximations and integrate them. Let us consider a channel $L \in Channels(\nu)$ with a describing function f_L and the point set $f_L([0, 1])$. The *Length* operator is defined as

$$Length(L) = \int_{f_L(0)}^{f_L(1)} \sqrt{1 + (\partial f_L(x) / \partial x)^2} dx$$

This method may also be used to calculate the length between any two points in the same channel. Here we simply integrate from the first point in the channel to the second point in the channel. Consider again a channel L as described above, and the two points $p = f_L(a)$ and $q = f_L(b)$

with $a, b \in [0, 1]$. We define a modified *Length* operator with the signature $Length : [A] \times [0, 1] \times [0, 1] \rightarrow \mathbb{R}$ as

$$Length(L, a, b) = \int_{f_L(a)}^{f_L(b)} \sqrt{1 + (\partial f_L(x)/\partial x)^2} \partial x$$

Another variation of the *length* operator is an extension of the first version with the same signature $Length : [A] \rightarrow \mathbb{R}$. But in this case, the *length* operator takes a complete spatial network $\nu : [A]$ as argument and sums up the lengths of all the channels in the spatial network. It is defined as

$$Length(\nu) = \sum_{L \in Channels(\nu)} Length(L)$$

4.3 Shortest Route

One of the classical queries in a spatial network is the shortest route (path) query. The task is to find a route between two points in a network which has the least distance among all routes between the two points. Shortest route queries are used to automatically find driving directions between physical locations, e.g., between two cities. The *ShortestRoute* operator finds such a shortest route between two points $p, q \in L(\nu)$ in a network $\nu : [A]$. The signature of this operator is $ShortestRoute : [A] \times \mathbb{R} \times \mathbb{R} \rightarrow [A]$. Its definition is given as

$$ShortestRoute(\nu, p, q) = \{sr \mid \begin{array}{l} \text{(i) } sr \in Routes(\nu, p, q) \\ \text{(ii) } \forall r \in Routes(\nu, p, q) : Length(r) \geq Length(sr) \end{array}\}$$

This operator checks all the routes between p and q and compares their length. It chooses the route with the smallest length as the shortest route. Since there could be several shortest paths, it returns all of them in a set.

5. SPATIAL NETWORK QUERY LANGUAGE

In this section, we introduce an SQL-like query language for spatial networks. We call the language *Spatial Networks Query Language (SNQL)*. We assume that a database has been created which holds spatial networks natively in it. Consider a national highway system represented as a spatial network. A traveler wants to drive from Gainesville to Atlanta. The main aim of the traveler is to reach Atlanta in the least amount of time. This means that he would like to travel on the route with the shortest length. Assuming we have a network containing the national highways, this query may be formulated as

```
select ShortestRoute(Gainesville, Atlanta) as sr
from NationalHighway
```

Sometimes the shortest route may not necessarily be the least time taking route. There might be congestions and other causes of delay along this route. Hence, a traveler might be more interested in having a set of possible routes from Gainesville to Atlanta. He may then choose his preferred route based on various other considerations like speed limits and congestions. This may be formulated in a query. But the number of possible paths from Gainesville to Atlanta might be large. Thus the query should have a limit on the number of routes it will return. In this particular case, we restrict the network distance of the paths to not more than 500 miles. The query is as follows:

```
select Routes(Gainesville, Atlanta) as sr
from NationalHighway
where Length(sr) < 500
```

In the next query, the select clause is used to project out a particular label attribute from the network. For example, a query to find the average speed of the route from Gainesville to Atlanta could be formulated as follows:

```
select avg(speed_limit)
from (select ShortestRoute(Gainesville, Atlanta)
from NationalHighway)
```

This query finds the shortest route between Gainesville and Atlanta from the National Highway network in the inner query, and then uses the aggregate function *avg* to calculate the average of all values of the label attribute *speed_limit* with respect to the determined route. The label attribute is assumed to belong to the *theme_attr* part (see Section 3.1) of the label type of the network *NationalHighway*.

6. CONCLUSIONS AND FUTURE WORK

This paper introduces and provides an overview of a formal data model for spatial networks that serves as a specification for a later implementation and integration in spatial (network) databases, Geographic Information Systems, transportation systems, and navigation systems. The data model is accompanied by some common operations, and the Spatial Networks Query Language (SNQL) demonstrates how the operations may be used in a database context.

This paper is the beginning of a larger effort to create a *Spatial Networks Algebra (SNA)* that can model a broad range of spatial networks and will have a comprehensive collection of operations and predicates defined on them. The *Spatial Networks Algebra* will later be combined with *spatial partitions* to create a *Map Algebra*.

7. REFERENCES

- [1] M. Erwig and R. H. Güting. Explicit Graphs in a Functional Model for Spatial Databases. *IEEE Transactions on Knowledge and Data Engineering*, 6(5):787–804, 1994.
- [2] T. Fleming and B. Mellor. An Introduction to Virtual Spatial Graph Theory. In *Int. Workshop on Knot Theory for Scientific Objects*, 2007.
- [3] H. Güting, T. de Almeida, and Z. Ding. Modeling and Querying Moving Objects in Networks. *The VLDB Journal*, 15(2):165–190, 2006.
- [4] R. Güting. GraphDB: Modeling and Querying Graphs in Databases. *20th Int. Conf. on Very Large Databases*, pages 297–308, 1994.
- [5] C. S. Jensen, T. B. Pedersen, L. Speicys, and I. Timko. Data Modeling for Mobile Services in the Real World. In *Advances in Spatial and Temporal Databases*, pages 1–9. Springer-Verlag, 2003.
- [6] S. Scheider and W. Kuhn. Road Networks and Their Incomplete Representation by Network Data Models. In *5th Int. Conf. on Geographic Information Science (GIScience)*, pages 290–307. Springer-Verlag, 2008.