Uncertainty Management for Network Constrained Moving Objects*

Zhiming Ding and Ralf Hartmut Güting

Praktische Informatik IV
Fernuniversität Hagen, D-58084 Hagen, Germany
{zhiming.ding, rhg}@fernuni-hagen.de

Abstract. One of the key research issues with moving objects databases (MOD) is the uncertainty management problem. In this paper, we discuss how the uncertainty of network constrained moving objects can be reduced by using reasonable modeling methods and location update policies. Besides, we present a framework to support variable accuracies in presenting the locations of moving objects. The operation design issues with uncertainty involved are also discussed.

1. Introduction

Uncertainty management is an important research issue in the moving objects database (MOD) technology. As stated in [8, 13], due to network delays, measuring errors, and the limitations of sampling methods, uncertainty is an inherent aspect of MOD.

The uncertainty management problem in MOD is actually two-fold – first, how to reduce uncertainty through reasonable modeling methods and location update policies; and second, how to express uncertainty and deal with uncertain data in databases by introducing suitable data types, operations, and index structures.

In recent years, a lot of research has been focused on the uncertainty management problem for moving objects, especially on the second aspect of the problem. In [8], Pfoser and Jensen *et al.* have analyzed the sources of uncertainty in presenting the locations of moving objects, and a representation framework has been proposed to deal with uncertain data. The work in [9] has explored the uncertainty and fuzziness in managing moving objects, and a mechanism is provided to deal with temporal, spatial, and spatio-temporal indeterminacies. In [12, 13], Trajcevski and Wolfson *et al.* have discussed the uncertainty management strategies in the DOMINO system. By applying an uncertainty threshold, the trajectory of a moving object is extended from a curve to a tube in the X×Y×T space and the operations, such as **inside**, are extended by introducing the uncertainty semantics such as "sometimes", "always", "possibly", and "definitely". In [11], a set of data types and operations have been proposed for the uncertainty management of moving objects. However, nearly none of these works have treated the interactions between moving objects and the underlying transportation networks in anyway. The works in [4, 7, 10] have discussed the uncertainty man-

* This research was supported by the Deutsche Forschungsgemeinschaft (DFG) research project "Databases for Moving Objects" under grant number Gu293/8-1.

agement problems with transportation networks involved. However, transportation networks are only used to reduce sampling noises from GPS or to predict future positions of moving objects.

In this paper, we discuss how the uncertainty of network constrained moving objects can be reduced through reasonable modeling methods and location update policies. We will mainly focus on the uncertainty caused by sampling methods so that we assume the uncertainty caused by other factors to be negligible.

The remaining part of this paper is organized as follows. Section 2 describes the basic methodologies in managing the uncertainty of network constrained moving objects; Section 3 deals with variable granularities in uncertainty management; Section 4 analyzes operation design issues with uncertainty involved; and Section 5 finally concludes the paper.

2. Basic Uncertainty Management Strategies for Network Constrained Moving Objects

In this section, we first present the basic data model for network constrained moving objects and the corresponding location update policies, and then we analyze the uncertainty problem under this framework.

2.1 The Basic Data Model to Present Network Constrained Moving Objects

Let's first deal with the underlying transportation networks. For simplicity, we model the whole transportation network as one single graph (for a more complete model, see [3]), and we will use "transportation network" and "transportation graph" interchangeably throughout this paper.

Definition 1 (transportation graph) A transportation graph G is defined as a pair:

$$G = (R, J)$$

where R is a set of routes and J is a set of junctions.

Definition 2 (route) A route of graph G, denote by r, is defined as follows:

$$r = (rid, route, len, fdr)$$

where *rid* is the identifier of r, *route* is a polyline which describes the geometry of r, *len* is the length of r, and $fdr \in \{0, 1, 2\}$ is the direction of the traffic flow allowed in r.

The polyline *route* in the above definition can be defined as a series of points in the Euclidean space. For simplicity, we suppose that the graph is spatially embedded in the X×Y plane so that the polyline can be presented as a series of points of the (x, y) form. The polyline is considered directed, which enables us to speak of the beginning point (or 0-end) and the end point (or 1-end) of the route.

The direction of traffic flow allowed in a route can have three possibilities, which are specified by *fdr*, whose value can assume 0, 1, 2, which corresponds to "from 0-end to 1-end", "from 1-end to 0-end", and "both directions allowed" respectively.

Definition 3 (junction) A junction of graph G, denoted by j, is defined as follows:

$$j = (jid, loc, ((rid_i, pos_i))_{i=1}^{n}, m)$$

where *jid* is the identifier of *j*, *loc* is the location of *j* which can be represented by a point value in the X×Y plane, *m* is the connectivity matrix [3] of *j* and $((rid_i, pos_i))_{i=1}^{n}$ describes the routes connected by *j*, where rid_i is the identifier of the *i*th route and $pos_i \in [0, 1]$ describes the position of the junction in the *i*th route. We suppose that the total length of any route is 1, and then every location in the route can be presented by a real number $p \in [0, 1]$.

Based on the above definitions for transportation networks, we can then define some useful data types – graph point, graph route section, and graph region, which form the basis for the modeling and querying of moving objects.

Let junct(*jid*), route(*rid*) be functions which return the junction and the route corresponding to the specified identifiers respectively. Let juncts(*G*) and routes(*G*) be the set of junctions and the set of routes of *G* respectively.

Definition 4 (graph point) A graph point is a point residing in the graph. The set of graph points of graph *G*, denoted by *GP*, is defined as follows:

$GP = \{(rid, pos) \mid \text{route}(rid) \in \text{routes}(G), \text{ and } pos \in [0, 1]\}$

In this definition, we only cite the route information since the locations of junctions can be represented by the locations in routes.

Definition 5 (graph route section) A graph route section is a part of a route. The set of graph route sections of graph *G*, denoted by *GRS*, can be defined as follows:

$GRS = \{(rid, S) \mid \text{route}(rid) \in \text{routes}(G), \text{ and } S \subseteq [0, 1]\}$

Definition 6 (graph region) A graph region is defined as a set of junctions and a set of route sections. The set of graph regions of graph *G*, denoted by *GR*, is defined as follows:

$$GR = \{(V, W) \mid V \subseteq GJ, W \subseteq GRS\}$$

where $GJ = \{jid \mid \text{junct}(jid) \in \text{juncts}(G)\}$.

Based on the above definitions of network constrained data types, we can then model network constrained moving objects. Since in most cases a moving object can be viewed as a point, moving objects are modeled as moving graph points. A moving graph point, *mgp*, is a function from time to graph point, that is:

$$mgp = f: T \rightarrow GP$$

where *T* is the domain of time, and *GP* is the domain of graph point of the graph.

In implementation, this function should be translated into a discrete representation. That is, a moving graph point is expressed as a set of moving units, and each moving unit describes one single moving pattern of the moving object for a certain period of time.

Definition 7 (moving graph point) a moving graph point, *mgp*, is defined as a sequence:

$$mgp = (t_i, (rid_i, pos_i), vm_i)_{i=1}^{n}$$

where t_i is a time instant, $(rid_i, pos_i) = gp_i$ is a graph point describing the location of the moving object at time t_i, and vm_i is the speed measure of the moving object at time t_i. We call $(t_i, (rid_i, pos_i), vm_i) = \mu_i$ the *i*th “moving unit” of *mgp*. For a running moving object, its last moving unit, μ_n, contains predicted information so that it is called “active moving unit”, which contains key information for location update algorithms (see Subsection 2.2).

The speed measure vm is a real number value. Its abstract value is equal to the speed of the moving object, while its sign (either positive or negative) depends on the direction of the moving object. If the moving object is moving from 0-end towards 1-end, then the sign is positive. Otherwise, if it is moving from 1-end to 0-end, the sign is negative.

For a valid moving graph point value, the following conditions should be met:

(1) $\forall i \in \{1, \ldots n\text{-}1\}$: ($rid_i = rid_{i+1}$) $\vee$ (gp_{i+1} geographically coincides with a junction which connects gp_i and gp_{i+1});

(2) $\forall i \in \{1, \ldots n\text{-}1\}$: viable(gp_i, gp_{i+1});

(3) $\forall i \in \{1, \ldots n\text{-}1\}$: $t_i < t_{i+1}$, and the moving object is assumed to move at even speed between t_i and t_{i+1}.

In the above definition, viable(gp_i, gp_{i+1}) means that through route(rid_i) or the junction which connects gp_i and gp_{i+1}, moving objects can transfer from gp_i to gp_{i+1}.

The main benefit of modeling moving objects on networks is that inside a certain route, the movement of the moving object is reduced to 1-demensional so that the index structures, the location update policies, and the related uncertainty management issues can be simplified.

2.2 The Location Update Policy for Network Constrained Moving Objects

In Definition 7, we assume that between two consecutive moving units, the moving object moves at even speed so that the position of the moving object at any time instant is a precise graph point. However, this is only an ideal situation. In real-world MOD applications, moving units are generated by location updates. If every speed change corresponds to a location update, the communication and computation cost can be too much expensive. To balance precision and communication costs, the system typically uses some predefined threshold to trigger location updates [14] so that the moving object between two consecutive location updates only moves at roughly even speed, which yields uncertainty (see Subsection 2.3).

In the network constrained MOD system, we suppose that every moving object is equipped with a portable computing platform and a GPS. The GPS enables the moving object to measure its current position in the (x, y) form (suppose that the MOD application is based on X×Y plane). With the support of some algorithms, the computing platform of the moving object can transform this location information to the (rid, pos) form, where rid is the identifier of the route where the moving object is located, and pos $\in$ [0, 1] is the location of the moving object inside the route.

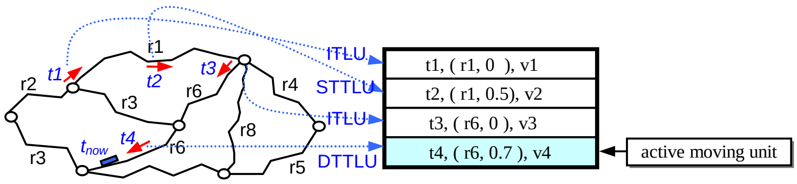
When a moving object mo initiates its journey in the MOD system, it needs to send to the server a location update message of the following form:

$msgu = (mid, t_u, (rid_u, pos_u), vm_u)$

where mid is the identifier of mo, t_u is the time when the location update is triggered, (rid_u, pos_u) = gp_u is the location (expressed as a graph point) of mo at time t_u, and vm_u is the speed measure of mo at time t_u.

Whenever receiving a location update message, the server will simply extract the information contained in it and generate a corresponding moving unit. This moving unit is then appended to the moving graph point value of the moving object as the active moving unit. The moving object will also keep the active moving unit for location update purposes.

During its move, the moving object will compare its actual location with the computed location derived from the active moving unit. Whenever certain conditions are met, a location update will be triggered. In the network constrained MOD system, there are three kinds of location updates – the ID-Triggered Locations Update (ITLU), the Distance-Threshold-Triggered Location Update (DTTLU), and the Speed-Threshold-Triggered Location Update (STTLU), as shown in Figure 1.



a) journey of a moving object b) corresponding moving unites

Fig. 1. An example moving graph point value

(1) ITLU. Whenever the moving object changes from one route to another via a junction, a location update will be triggered, and a new moving unit corresponding to the entering point to the new route will be appended to the corresponding moving graph point value of the moving object.

(2) DTTLU. A location update will be triggered whenever the difference between the computed position derived from the active moving unit and the actual position measured by the milemeter exceed a certain predefined distance threshold ξ. The computed location of the moving object is as follows (suppose the active moving unit is $(t_n, (rid_n, pos_n), vm_n)$):

$$pos_{now} = pos_n + vm_n \times (t_{now} - t_n)$$

(3) STTLU. Whenever the difference between the current speed of the moving object and the speed recorded in the active moving unit exceeds a certain predefined speed threshold ψ, a location update will be triggered.

2.3 Computing the Locations of Moving Objects with Uncertainty Considered

As stated in Subsection 2.2, since inside a moving unit, a moving object moves only roughly at even speed, its location is no longer a precise graph point at any time instant. Actually, its position other than the location update times becomes uncertain, and we can only describe its position by the concept of "possible location". In this subsection, we discuss how the possible location of the moving object at any given time instant can be computed through its moving unit and the thresholds predefined in the system.

We suppose that the corresponding moving units of an active moving object, *mo*, is as follows:

$$mgp = ((t_i, (rid_i, pos_i), vm_i))_{i=1}^{n}$$

Let $\mu_i = (t_i, (rid_i, pos_i), vm_i)$ $(1 \le i \le n)$ be the ith moving unit of the moving object, and $gp_i = (rid_i, pos_i)$ be the location of the moving object at time t_i. For the sake of simplicity, we assume that the speed measure vm_i is positive, which means that the moving object is moving from 0-end towards 1-end along route(rid_i). The methodology can be easily adapted to the situation when the speed measure is negative.

Let $v^i_{\max} = vm_i + \psi$ and $v^i_{\min} = vm_i - \psi$, where ψ is the speed threshold. In the following discussion, we suppose that ξ, $v^i_{\min}$, and $v^i_{\max}$ have already been transformed to the [0, 1] scope according to the length of the corresponding route.

Case 1. $\exists\, i \in \{1, \ldots, n\} : t_q = t_i$

In this case, t_q happens to be a location update time, and the possible location of the moving object is a precise graph point $gp_q = (gid_i, rid_i, pos_i)$.

Case 2. $\exists\, i \in \{1, \ldots, n-1\} : t_i < t_q < t_{i+1}$

If $rid_i = rid_{i+1}$, we can be assured that the moving object is on route(rid_i) at time t_q. Suppose its position in route(rid_i) is $p_q \in [0, 1]$. p_q satisfies the following conditions:

1) $pos^*_q - \xi \le p_q \le pos^*_q + \xi$, where $pos^*_q = pos_i + vm_i \times (t_q - t_i)$. Otherwise there would be a DTTLU between t_i and t_{i+1};
2) $pos_i + v^i_{\min} \times (t_q - t_i) \le p_q \le pos_i + v^i_{\max} \times (t_q - t_i)$. Otherwise there would be an STTLU between t_i and t_{i+1};
3) $pos_{i+1} - v^i_{\max} \times (t_{i+1} - t_q) \le p_q \le pos_{i+1} - v^i_{\min} \times (t_{i+1} - t_q)$. Otherwise, the moving object would not be able to arrive at gp_{i+1} in time without triggering an STTLU.

Therefore, the possible location of the moving object at time t_q is a graph route section $grs_q = (gid_n, rid_n, [p_{qmin}, p_{qmax}])$, where p_{qmin}, p_{qmax} can be computed in the following way:

$$p_{qmin} = \max(0,\ pos^*_q - \xi,\ pos_i + v^i_{\min} \times (t_q - t_i),\ pos_{i+1} - v^i_{\max} \times (t_{i+1} - t_q))$$

$$p_{qmax} = \min(1,\ pos^*_q + \xi,\ pos_i + v^i_{\max} \times (t_q - t_i),\ pos_{i+1} - v^i_{\min} \times (t_{i+1} - t_q))$$

where $pos^*_q = pos_i + vm_i \times (t_q - t_i)$.

If $rid_i \ne rid_{i+1}$, then we know that μ_{i+1} is generated by an ITLU, and gp_{i+1} geographically coincides with a junction. In this case, before proceeding with the above procedure, we need first to transform gp_{i+1} to the corresponding graph point in route(rid_i).

Case 3. $t_n < t_q \le t_{now}$

In this case, we know that the moving object is still on route(rid_n), and its location in route(rid_n) at time t_q, denoted by p_q ($p_q \in [0,1]$), satisfies the following conditions:

1) $pos^\Diamond_q - \xi \le p_q \le pos^\Diamond_q + \xi$, where $pos^\Diamond_q = pos_n + vm_n \times (t_q - t_n)$. Otherwise there would be a DTTLU triggered after t_n;
2) $pos_n + v^n_{\min} \times (t_q - t_n) \le p_q \le pos_n + v^n_{\max} \times (t_q - t_n)$. Otherwise there would be a STTLU triggered after t_n;

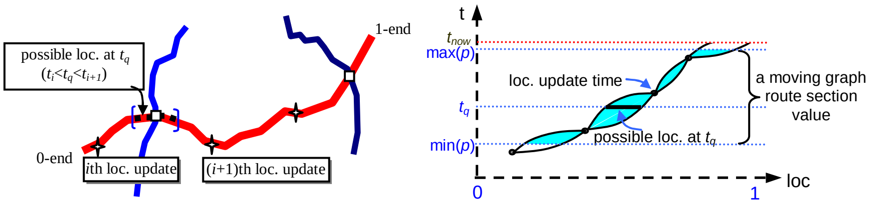
Therefore, the possible location of the moving object at time t_q is a graph route section $grs_q = (gid_n, rid_n, [p_{qmin}, p_{qmax}])$, where p_{qmin}, p_{qmax} can be computed as follows:

p_{qmin} = max (0, $pos^{\lozenge}_{q} - \xi$, $pos_n + v^{n}_{\min} \times (t_q - t_n)$)

p_{qmax} = min (1, $pos^{\lozenge}_{q} + \xi$, $pos_n + v^{n}_{\max} \times (t_q - t_n)$)

where $pos^{\lozenge}_{q} = pos_n + vm_n \times (t_q - t_n)$.

As described above, the possible location of the moving object is a moving graph route section, as shown in Figure 2 (for simplicity, we only depict the situation when the moving objects is moving inside one route. For the whole trajectory covering multiple routes, see Figure 4).



a) possible location of a moving object   b) corresponding moving graph route section value

Fig. 2. Possible location of a moving object

As illustrated in Figure 2, when uncertainty is considered, the possible location of the moving object at any time instant can be computed, which is a graph route section. This is a main advantage compared with the model of moving objects on free X×Y plane, where the possible location of a moving object is a region. To generalize the concept of moving graph route section for the design of operators, we define it as follows.

Definition 8 (moving graph route section) a moving graph route section, *mgrs*, is defined as the following form:

$$mgrs = ((t_i, (rid_i, pos_i), vm_i)\ ^{n}_{i=1}, p)$$

where $(t_i, (rid_i, pos_i), vm_i)\ ^{n}_{i=1}$ includes all the moving units of the moving object generated by location updates, and p is a period value which is composed of a set of disjoint time intervals. Figure 2(b) illustrates an example moving graph route section value (the shaded part).

For a valid moving graph route section value, the following conditions should be met (suppose $gp_i = (rid_i, pos_i)$):

(1) $\forall i \in \{1, \ldots n\text{-}1\}$: $(rid_i = rid_{i+1}) \vee (gp_{i+1}$ geographically coincides with a junction which connects gp_i and $gp_{i+1})$;

(2) $\forall i \in \{1, \ldots n\text{-}1\}$: viable(gp_i, gp_{i+1});

(3) $t_1 \le \min(p) < \max(p) \le t_{now}$ (if the moving object is still active in the system)
$t_1 \le \min(p) < \max(p) \le t_n$ (if the moving object has finished its trip)

3. Multiple Granularities in Uncertainty Management

By adjusting the distance threshold ξ and the speed threshold ψ, the framework presented in Section 2 can support variable precisions in presenting the locations of moving objects. However, sometimes this is still "too precise". In a lot of cases, much lower precisions in presenting the locations of moving objects, for instance, "from 8:00 to 10:00, I was traveling in the city center; after that until 13:00 I was visiting the museum area; and from 13:00 to 15:00 I was in the university area" is quite acceptable.

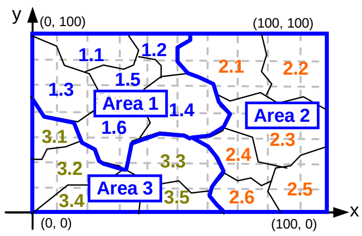
To better support variable precisions in presenting the locations of moving objects, we define a new data type, discretely moving graph region, to describe the possible locations of moving objects

Definition 9 (discretely moving graph region) A discretely moving graph region is defined as a sequence of the following form:

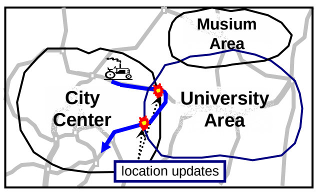
$$dmgr = ((t_i, gregion_i))_{i=1}^{n}$$

where t_i is a time instant, $gregion_i$ is a graph region value. For $\forall i \in \{1, \ldots n\text{-}1\}$, $t_i < t_{i+1}$, and the moving object is assumed to move inside $gregion_i$ between t_i and t_{i+1}. For simplicity, we also call $((t_i, gregion_i)$ $(1 \le i \le n)$ a "moving unit" of the moving object.

The locations of moving objects can be tracked in the following way. First, the whole transportation network is partitioned into a group of areas with each area to be a graph region. To support multiple granularities in uncertainty management, the system can have multiple partitions on the same traffic network, which form a hierachical structure (as shown in Figure 3(a)). The graph regions are uniquely numbered, and both the server and the moving object need to keep the partition information. To avoid frequent location updates when moving objects are moving near the border between two partitions, these partition areas should overlap each other to some extent, as shown in Figure 3(b).



a) partition of traffic network



b) overlap of partition areas

Fig. 3. Partition of the underlying traffic network

When a moving object starts its trip, it needs to send to the server its current region number. During its life time, whenever it crosses the border of the current region, a location update will be triggered. The location update message includes the time and the number of the new graph region where the moving object is located. When the server receives a location update message, it will simply generate a moving unit and append it to the discretely moving graph region value of the moving object.

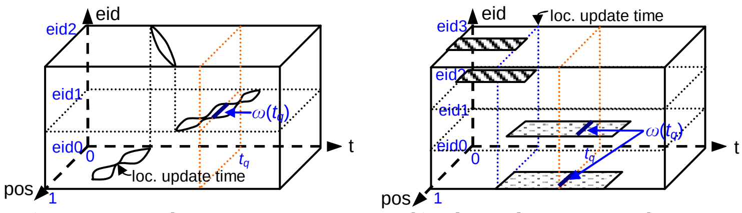
Under this framework, the possible location of a moving object at any given time instant t_q is a graph region. If $t_i \le t_q < t_{i+1}$, then the possible location of the moving object is $gregion_i$. If $t_n \le t_q < t_{now}$, then the possible location is $gregion_n$.

By selecting a suitable granularity, the moving object can control the communication and computation costs caused by location updates.

4. Taking Uncertainty into the Design of Operations

In [5], Güting *et al*. have defined a rich set of operations for network constrained moving objects and the related data types. Through some extension, these operations can be upgraded to support uncertainty. In this section, we do not aim to present a full design of operations with uncertainty involved. Instead, we only outline some general ideas behind the design.

First let's see the "trajectories" of moving objects. If a moving object is modeled directly in the Euclidean space, its trajectory is a curve (or a tube when uncertainty is considered) in the X×Y×T space [8, 13]. However, in network constrained moving objects databases, the trajectory of a moving object can have totally different forms, as shown in Figure 4.



a) a moving graph route section b) a discretely moving graph region

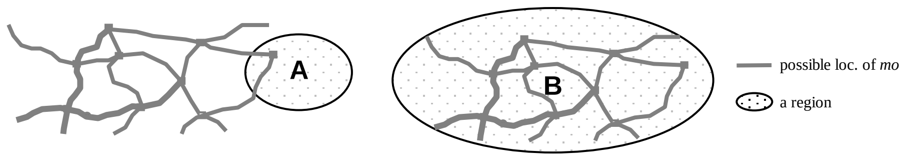
Fig. 4. "Trajectories" of moving objects

As illustrated in Figure 4, the possible location of a moving object at any given time instant t_q, denoted by $\omega(t_q)$, is the intersection of its trajectory and the plane corresponding to t_q (which is vertical to the t–axis). $\omega(t_q)$ can be either a graph route section (see Figure 4(a)), or a graph region (see Figure 4(b)).

Let $\partial\omega(t_q)$ be the set of points contained in $\omega(t_q)$. In designing an operation which involves the whole trajectory or part of the trajectory, say **inside**, we need to extend it to **inside_possibly** and **inside_definitely** with the following semantics:

inside_possibly($\omega(t_q), A$) ⇔ $\exists p \in \partial\omega(t_q)$: **inside**(p, A) ⇔ **intersects**($\omega(t_q), A$)
inside_definately($\omega(t_q), B$) ⇔ $\forall p \in \partial\omega(t_q)$: **inside**(p, B) ⇔ **inside**($\omega(t_q), B$)

Figure 5 illustrates the semantics of the extended **inside** operation.



a) inside_possibly b) inside_definately

Fig. 5. Semantics of **inside_possibly** and **inside_definately**

Following the strategy described above, we can extend other operations, such as **at**, **intersect**, **atperiod**, and **atinstant,** with uncertainty involved. For instance, the signature of the **atperiod** and **atinstant** operations can be extended as follows (<u>*mgrs*</u>, <u>*dmgr*</u>, <u>*grs*</u>, <u>*gr*</u> are the data types corresponding to moving graph route section, discretely moving graph region, graph route section, and graph region respectively):

atperiod: <u>*mgrs*</u> × <u>*period*</u> → <u>*mgrs*</u>
<u>*dmgr*</u> × <u>*period*</u> → <u>*dmgr*</u>

atinstant: <u>*mgrs*</u> × <u>*instant*</u> → <u>*grs*</u>
<u>*dmgr*</u> × <u>*instant*</u> → <u>*gregion*</u>

5. Conclusion

One of the key research issues with moving objects databases (MOD) is the uncertainty management problem. In this paper, we have discussed how the uncertainty of network constrained moving objects can be reduced by using reasonable modeling methods and location update policies. Besides, we have presented a framework to support variable accuracies in presenting the locations of moving objects. The operation design issues with uncertainty involved have also been discussed.

The above mechanism is designed for implementation in the Secondo system [1]. We have designed a rich set of data types and operations for moving objects and the underlying transportation graphs [5, 2], which are under development as three algebra modules, spatial algebra, transportation graph algebra, and moving object algebra, in the Secondo system.

The future research includes the full design of the operations with uncertainty considered, the index structures for the uncertain trajectories of the network constrained moving objects, and the corresponding query processing techniques.

References

[1] Dieker S, Güting R H. Plug and Play with Query Algebras: SECONDO. A Generic DBMS Development Environment. *Proc. of IDEAS 2000*, Yokohoma, Japan, 2000.

[2] Ding Z, Güting R H. Modeling Temporally Variable Transportation Networks. *Proc. of DASFAA'2004*, Jeju Island, Korea, March 2004.

[3] Ding Z, Güting R H. Managing Moving Objects on Dynamic Transportation Networks. *Proc. of SSDBM'2004*, Santorini, Greece, June 2004.

[4] Gowrisankar H, Nitte S. Reducing Uncertainty in Location Prediction of Moving Objects in Road Networks. *Proc. of GIScience 2002*, Colorado, 2002.

[5] Güting R H, Almeida V T, Ding Z. Modeling and Querying Moving Objects in Networks, Technical Report 308, Fernuniversität Hagen, 2004.

[6] Güting R H, Böhlen M H, Erwig M, Jensen C S, *et al*. A Foundation for Representing and Querying Moving Objects. *ACM Transactions on Database Systems*, 25(1), 2000.

[7] Meratnia N, Kainz W, *et al*. Spatio-temporal Methods to Reduce Data Uncertainty in Restricted Movement on a Road Network, *Proc. of joint ISPRS, IGU and CIG Symp.on Geospatial theory,Processing and Applications*, Canada, 2002.

[8] Pfoser D, Jensen C S. Capturing the Uncertainty of Moving Object Representations. *Proc. of SSD'99*, Hong Kong, China, July, 1999.

[9] Pfoser D, Tryfona N. Capturing Fuzziness and Uncertainty of Spatiotemporal Objects. *ADBIS 2001*, Vilnius, Lithuania, Sept. 2001

[10] Schlenoff C, Madhavan R, Balakirsky S. An Approach to Predicting the Location of Moving Objects during On-Road Navigation. *IJCAI'2003*, Mexico, August, 2003.

[11] Tøssebro E, Nygård M. Uncertainty in Spatiotemporal Databases, *Proc. 2nd Biennial Int. Conf. on Advances in Information Systems (ADVIS)*, Turkey, Oct., 2002.

[12] Trajcevski G, Wolfson O, Cao H, Lin H, Zhang F, Rishe N. Managing Uncertain Trajectories of Moving Objects with Domino. *ICEIS'02*, Spain, April, 2002

[13] Trajcevski G, Wolfson O, Chamberlain S, Zhang F. The Geometry of Uncertainty in Moving Objects Databases, *Proc. of EDBT'02*, Prague, Czech Republic, 2002.

[14] Wolfson O, Xu B, Chamberlain S, Jiang L. Moving Object Databases: Issues and Solutions. *Proc. of SSDBM'98*, Capri, Italy, July 1998.