Querying Moving Objects with Uncertainty in Spatio-temporal Databases

Hechen Liu & Markus Schneider*

Department of Computer and Information Science and Engineering
University of Florida
Gainesville, FL 32611, USA
{heliu, mschneid}@cise.ufl.edu

Abstract. Spatio-temporal uncertainty is a special feature of moving objects due to the inability of precisely capturing or predicting their continuously changing locations. Indeterminate locations of moving objects at time instants add uncertainty to their topological relationships. Spatio-temporal uncertainty is important in many applications, for example, to determine whether two moving objects could possibly meet. Previous approaches, such as the 3D cylinder model and the space-time prism model have been proposed to study the spatio-temporal uncertainty. However, topological relationships between uncertain moving objects have been rarely studied and defined formally. In this paper, we propose a model called *pendant model*, which captures the uncertainty of moving objects and represents it in a databases context. As an important part of this model, we define a concept called *spatio-temporal uncertainty predicate* (*STUP*) which expresses the development of topological relationships between moving objects with uncertainty as a binary predicate. The benefit of this approach is that the predicates can be used as selection conditions in query languages and integrated into databases. We show their use by query examples. We also give an efficient algorithm to compute an important STUP.

1 Introduction

The study of moving objects has aroused a lot of interest in many fields such as mobile networking, transportation management, weather report and forecasting, etc. Moving objects describe the continuous evolution of spatial objects over time. The feature that their locations change continuously with time makes them more complicated than static spatial objects in some aspects; one aspect refers to the topological relationships. A topological relationship, such as *meet*, *disjoint* or *inside*, characterizes the relative position between two or more spatial objects. In the spatio-temporal context, however, topological relationships between moving objects are not constant but may vary from time to time. For

* This work was partially supported by the National Science Foundation (NSF) under the grant number NSF-IIS-0812194 and by the National Aeronautics and Space Administration (NASA) under the grant number NASA-AIST-08-0081.

example, an airplane is disjoint with a hurricane at the beginning, and later it flies approaching the hurricane, and finally locates inside the hurricane. This developing relationship is named as *enter* [1]. Since we often do not have the capability of tracking and storing the continuous changes of locations all the time due to the deficiency of devices but can merely get observations of them at time instants, the movements between these time instants are always uncertain. The indeterminate locations further add uncertainty to the topological relationships between two moving objects.

Traditionally, moving objects are represented as 3D (2D+time) polylines. This representation, however, only stores the most possible trajectory of a moving object, without considering the uncertainty property. Several approaches have been proposed to handle spatio-temporal uncertainty, including the 3D cylinder model [2] and the space-time prism model [3]. However, the former assumes that the degree of uncertainty of a moving object does not change with time and thus lacks a precise representation. The latter is not able to combine both the certain part and the uncertain part of a movement. Both models rarely discuss the dynamic change of topological relationships with uncertainty.

The goal of this paper is to model and query the dynamic topological relationships of moving objects with uncertainty. This problem is solved through proposing the *pendant model* which is based on the well known space-time prism model while adds two significant advantages. First, it is an integrated and seamless model which combines both known movements and uncertain movements. Second, it formally defines spatio-temporal uncertainty predicates (STUP) that express the topological relationships between uncertain moving objects. This is important in querying moving objects with uncertainty since they can be used as selection conditions in databases. As an important part of the model, we formally define operations related to retrieving and manipulating uncertain data, and STUPs such as *possibly_meet_at*, *possibly_enter*, and *definitely_cross* that are defined on the basis of the operations. Queries related to the uncertainty in the topological relationships between moving objects can then be answered.

The paper is organized as follows: Section 2 discusses the related work on moving objects and spatio-temporal uncertainty models. Section 3 introduces our pendant model of moving objects with uncertainty. Section 4 defines spatio-temporal uncertainty predicates and shows the use of them in queries. Section 5 presents the algorithms to determine the defined spatio-temporal uncertainty predicates. Section 6 draws some conclusions and discusses future work.

2 Related Work

Several approaches have been proposed to model moving objects in spatial databases and GIS [4–6]. In some models, a moving object is represented as a polyline in the three-dimensional (2D+time) space with the assumption that the movement is a linear function between two consecutive sample points [4, 6]. The 3D polyline model, however, only yields the trajectory that the moving object may take with the highest possibility but is often not the exact route the moving object takes in reality.

An important approach that captures the uncertainty feature of movements is the *3D cylinder model* [7, 2]. A trajectory of a moving point is no longer a polyline but a cylinder in the 2D+time space. The possible location of a moving object at a time instant is within a disc representing the area of the uncertainty. The cylinder model, however, assumes that the degree of uncertainty does not change between two sample points, which is not the exact case in reality. An improved model, the space-time prism model [3] represents the uncertain movement of a moving object as a volume formed by the intersection of two half cones in the 3D space. Given the maximum speed of a moving object and two positions at the beginning and at the end of the movement, all possible trajectories between these two points are bounded within the volume of two half cones. Space-time prisms are more efficient than the cylinder models since they reduce the uncertain volume by two thirds because of the geometric properties of cones. There are many application examples that benefit from the space-time prism model [8–11]. In some GPS applications, the uncertainty is represented as as an error ellipse which is the projection of a space-time prism in 2D space [8]. The set of all possible locations is useful, for example, in determining whether an animal could have come into contact with a forest fire, or an airplane could have entered a hurricane [12]. An approach that uses the space-time prism model to provide an analytic solution to alibi queries is proposed in [13]. A most recent approach discusses the problem of efficient processing of spatio-temporal range queries for moving objects with uncertainty [14]. However, there are some remaining problems that are not solved by space-time prism-based approaches. First, some movements are more complicated since they include both known movements and unknown movements together, but this situation has not yet been included. Second, dynamic topological relationships between moving objects with uncertainty have not been discussed. We will provide solutions to the aforementioned problems in our model.

The spatio-temporal predicates (STP) model [1] represents the temporal development and change of the topological relationship between spatial objects over time. A spatio-temporal predicate between two moving objects is a temporal composition of topological predicates. For example, the predicate *Cross* is defined as $Cross := Disjoint - meet - Inside - meet - Disjoint$ where juxtaposition indicates temporal composition. However, the STP model does not deal with the uncertainty of topological relationships between moving objects.

3 Modeling the Uncertainty of Moving Objects

In this section, we introduce the pendant model which represents the uncertainty of moving objects. Section 3.1 provides the formalization of spatio-temporal uncertainty. We review the space-time prism model, and then taking it as a basis, we give the definition of a new concept called the uncertainty volume. Section 3.2 introduces the pendant model which integrates the known part of a movement and the uncertain part of movement together. Section 3.3 discusses operations on the pendant model.

3.1 The Formalization of Uncertain Movements

The pendant model is built on top of the well known space-time prism model of moving objects with uncertainty. Given the locations and time instants of the origin and the destination of a moving point and its maximum speed, all possible trajectories between these two time instants are bounded within the volume of two half cones. An advantage of using a cone to represent moving objects with uncertainty compared to a cylinder is that the former is one third of the latter in volume, which reduces the degree of uncertainty. We rewrite this relationship as a formal definition of the uncertainty volume of a moving point.

Definition 1 *Let (x_0, y_0) denote the origin of a moving point at time t_0, (x_1, y_1) denote the destination of this point at time t_1, where $t_1 > t_0$, and v_{max} denote the maximum velocity. Then the uncertainty volume UV is given as*

$$
\begin{aligned}
UV = \{\, (x, y, t) \mid \\
&\text{(i)}\ \ x, y, t \in \mathbb{R}, t_0 \leq t \leq t_1 \\
&\text{(ii)}\ \ t_0 \leq t \leq (t_0 + t_1)/2 \Rightarrow \sqrt{(x - x_0)^2 + (y - y_0)^2} \leq (t - t_0)v_{max} \\
&\text{(iii)}\ (t_0 + t_1)/2 \leq t \leq t_1 \Rightarrow \sqrt{(x - x_1)^2 + (y - y_1)^2} \leq (t_1 - t)v_{max} \}
\end{aligned}
$$

The above definition contains two different scenarios illustrated by Figure 1(a) and Figure 1(b) respectively. First, if the origin and the destination of the moving point are at the same location, the uncertain volume of the moving object is shown by two connected symmetric cones sharing a base parallel to the xy-plane, which is easy to prove. In Condition(ii), the right side of the inequality shows the maximum distance the moving point can travel up to a given time instant t, which is actually the radius of the circle representing the uncertainty area, and the left side represents the actual distance it has traveled. If we integrate all the time instants, the possible movements are bounded by the half cone with top (x_0, y_0). Condition (iii) shows that the moving point must return to its original location with the same speed bound. Thus, these two parts of movement form a symmetric double-cone. If we make a cut to the cone which is parallel to the xy-plane, we will get either a point or a circle object.

In the second scenario, the origin and the destination of the movement are different, in which case the uncertain volume is the combination of two oblique cones. The shared base of two cones is not parallel to the xy plane. The reason is that the moving point shows an apt direction from the origin to the destination, which is different from the first scenario. As Figure 1(b) illustrated, since $x_1 > x_2$, the moving point has an apt direction to the right of the plane. For the first half cone, the left most trajectory is shorter than the right most trajectory, which means that the moving point moves to left for a small distance, and suddenly changes its direction to the opposite and travels to the destination with a longer distance. Thus, the base of such an asymmetric double-cone is oblique. If we give a cut with a plane parallel to the xy plane on the oblique cone, we will get either a point or a lens object.

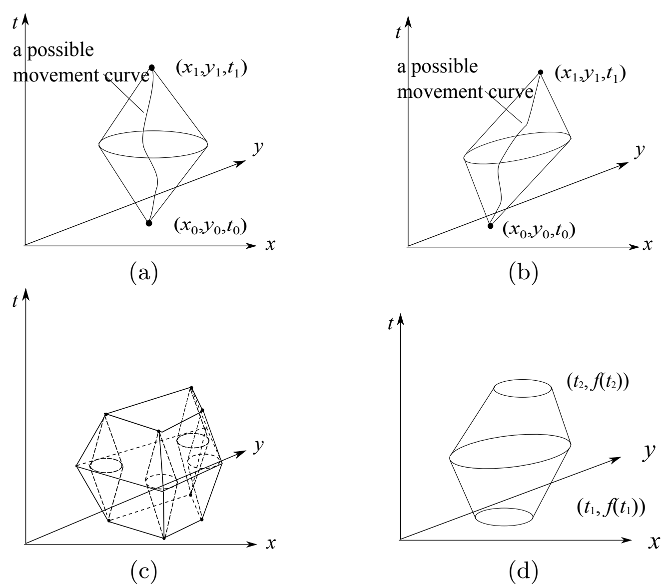


Fig. 1. Double-cone shaped volumes representing the uncertainty of a moving point with same (a) and different (b) origin and destination, uncertainty volumes of a moving polygon region (c) and a moving circle region (d).

We further extend Definition 1 to moving regions whose areas must be considered. We assume that a moving region is represented by a polygon (a circle region is a polygon with many vertices approaching infinite). Further we assume that a moving region only makes translation, i.e., there is no rotation, shrink or split of this region, then each vertices of the region is a moving point, and the uncertainty volume of each of them can be used to construct the uncertain volume of the entire moving region.

Definition 2 *Let $f(t)$ denote a moving region and let t_0 and t_1 denote two instants at the start and the end of the movement and $t_1 > t_0$, and v_{max} denote the maximum velocity. Then the uncertainty volume UV is defined as*

$$\begin{aligned}
UV = \{\,(x, y, t)\,|\, & \\
& \text{(i)}\;\; (x, y) \in f(t) \in region \\
& \text{(ii)}\;\; t_0 \le t \le (t_0 + t_1)/2 \Rightarrow \sqrt{(x - x_0)^2 + (y - y_0)^2} \le (t - t_0)v_{max} \\
& \text{(iii)}\;\; (t_0 + t_1)/2 \le t \le t_1 \Rightarrow \sqrt{(x - x_1)^2 + (y - y_1)^2} \le (t_1 - t)v_{max}\}
\end{aligned}$$

Condition (i) states that the uncertainty volume of the moving region contains all uncertainty volume of all points of the region. Condition (ii) and (iii) show that the uncertainty volume of each single point is a double-cone volume, as in Definition 1. Since the moving region only makes translation, all vertices will result the same uncertainty volume. Thus we can form the base of the entire uncertainty volume by connecting the tangent lines between bases of cones, as illustrated by Figure 1(c). In a special case, if the shape of the moving region is a circle, the uncertainty volume of it is a cone with circle shaped bases, as illustrated in Figure 1(d). We observe that Definition 1 can be treated as a special case of Definition 2 in that a moving point is a degenerated case of a moving polygon which contains only one vertex, thus we can replace (x_0, y_0) by $f(t_0)$ in the inequalities. In the rest of the paper, we will use the latter notation when treated moving points and moving regions together.

3.2 The Pendant Model as the Combination of Certain and Uncertain Movements

In some real cases, part of a movement can be exactly tracked, while part of the movement is uncertain. For example, the appearance of the signal on a radar representing the change of the position of an airplane belong to this category. A signal may exist in some intervals while it may disappear in some other intervals. The movements during periods in which the signal exists are known movements, however, the movements in the periods when the signal disappears are unknown but are interesting to us. We separate the movement of a moving object into two parts: the known part that can be represented as a function and is analogous to the string of a necklace, and the uncertain part which is analogous to a "pendant". It is allowed that in some special cases, a movement only contains known part, or only contains uncertain part, thus it will be convenient to give an integrated model that can represent all kinds of movement. In the above part, we have discussed the uncertain movement of a moving object in a single time interval. Next, we discuss how to integrate the known part of the movement. The part that is known can be represented by line segments through linear interpolation. This is different from the 3D polyline model, for the latter does not contain uncertainty parts. We say that a moving point, denoted by $(t, (x, y))$, is *linear* in $[t_0, t_1]$, if and only if

$$\forall t \in [t_0, t_1], \left(\frac{z - z_0}{t - t_0} = \frac{z_1 - z_0}{t_1 - t_0} \ \wedge \ z \in \{x, y\} \ \wedge \ x_1 \neq x_0, y_1 \neq y_0\right)$$
$$\vee \ (x = x_0 = x_1 \ \wedge \ y = y_0 = y_1)$$

Further, we say that a moving region, denoted by $f(t)$, is linear if and only if $\forall p \in f(t)$ is linear. A special case of the linear movement is that a static spatial object can be treated as a moving object whose locations remain constant over time. For example, a static point can be represented as a straight line perpendicular to the xy plane, and a static region can be represented as a cylinder volume in 2D+time space.

Definition 3 *The movement of a moving object with uncertainty, denoted by unmovement, is a function of a data type* α*, where* $\alpha \in \{point, region\}$*, defined as*

$unmovement = \{f : time \rightarrow \alpha \,|$
(i) $dom(f) = \cup_{i=1}^{n}[l_i, r_i], n, i \in \mathbb{N}; l_i, r_i \in time$
(ii) $\forall 1 \leq i < n : r_i \leq l_{i+1}$
(iii) $\forall 1 \leq i \leq n : f(t)$ *is continuous at* $t \in [l_i, r_i]$
(iv) *Let* $v = d(f(t))/d(t)$ *denote the velocity of* f *at* t,
$\forall 1 \leq i \leq n, \forall t \in [l_i, r_i] : v_{max_i} = max(v)\}$
(v) $\forall 1 \leq i \leq n, \forall t \in [l_i, r_i]$: *either* $(f(t), t)$ *is an uncertainty volume, or* $f(t)$ *is linear in* $[l_i, r_i]$.

Definition 3 gives the formal representation of the moving object data type with uncertainty in our pendant model. Condition(i) states that an *unmovement* is a partial function defined on a union of intervals. Condition(ii) ensures that time intervals do not overlap with each other thus maximizes intervals and gives a unique representation for a known movement. Condition (iii) states that the movement is continuous in each interval thus instantaneous jump is not allowed. Condition (iv) defines v_{max_i} as the upper bound of the speed of the moving object during each interval i. Condition (v) states that within each interval of the moving object function, the movement is either a certain movement which changes linearly, or an uncertain volume of a double-cone. Thus, the uncertainty volume of a moving object is composed by the union of cones and line segments.

On the basis of Definition 3, we further define two subclasses of the *unmovement* data type, unmpoint and unmregion, representing moving point with uncertainty and moving region with uncertainty, which inherit all properties of *unmovement*.

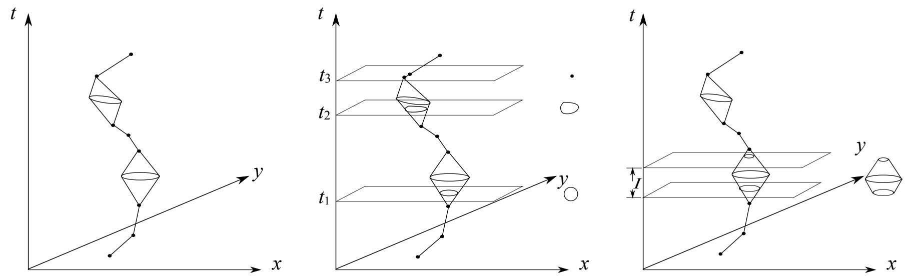
Definition 4 *The data types moving point with uncertainty, and moving region with uncertainty are defined as follows:*

$$unmpoint \subset unmovement = \{f : time \rightarrow \alpha \,|\, \exists t \in dom(f) : f(t) \in point\}$$
$$unmregion \subset unmovement = \{f : time \rightarrow \alpha \,|\, \forall t \in dom(f) : f(t) \in region\}$$

3.3 Operations on the Pendant Model

Operations are important components in a data model. They are integrated into databases and used as tools for retrieving and manipulating data. In the rest of this section, we introduce some important operations in our pendant model. They will be useful in helping us define spatio-temporal uncertain predicates in the next section. Because of the page limitation, we only give the semantic and the explanation of each operation here.

construct_segment	$: \alpha \times \alpha \times interval$	$\rightarrow segment$
construct_pendant	$: \alpha \times \alpha \times interval \times real$	$\rightarrow UV$
construct_movement	$: segment^m \times UV^n$	$\rightarrow unmovement$
at_instance	$: unmovement \times instant$	$\rightarrow \alpha$
temp_select	$: unmovement \times interval$	$\rightarrow unmovement$
dom	$: unmovement$	$\rightarrow interval$



(a) (b) (c)

Fig. 2. An *unmpoint* data object (a); at_instance (b) and temp_select (c)

The operation *construct_segment* takes two observations and an interval as inputs, to construct the certain movement, which is the a 3D line segment between the observations. The *construct_pendant* operation takes two observations, the speed of the moving object, and the interval as inputs, and construct a double-cone volume. The *construct_movement* operation integrates known movements and movements with uncertainty together, where $m, n \in \mathbb{N}$, which means that a movement is composed by multiple known parts and uncertain parts. The *at_instance* operation takes an *unmovement* data type and an instant as inputs, and returns a spatial object representing the uncertainty of the moving object at that instant, as illustrated in Figure 2(b). The *temp_select* operation is short for "temporal selection". It retrieves partial movement of the moving object during a time interval that is last for a period, as illustrated in Figure 2(c). The *dom* operator returns the life domain interval of an uncertain moving object.

4 Spatio-Temporal Predicates with Uncertainty and Queries

In this section, we discuss topological relationships between moving objects with uncertainty and show query examples. In [1], authors have defined topological relationships between moving objects over time as binary predicates, called *spatio-temporal predicates* (*STP*), the results of which are either true or false. This will make querying moving objects easier by using binary pre-defined predicates as selection conditions. Similarly, we define the dynmic topological relationships between moving objects with uncertainty as s*patio-temporal uncertain predicates* (*STUP*), which will be integrated into databases as join conditions. In Section 4.1, we formally define important STUPs on moving objects. In Section 4.2, we introduce the use of the STUP by showing query examples.

4.1 Definitions of Spatio-Temporal Uncertain Predicates

A spatio-temporal uncertain predicate (STUP) is a bool expression that is composed of topological predicates, math notations and operations on the pendant

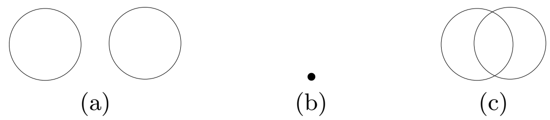


Fig. 3. Instant predicates of moving points with uncertainty: disjoint_at (a), definitely_meet_at and possibly_meet_at (c)

model we have defined in Section 3.3. An STUP expression may contain the distance operator *dist*, topological predicates between two regions (*disjoint*, *meet*, *overlap*, *covers*, *coveredBy*, *equal*, *inside*, *contains*), logic operators ($\neg$, $\exists$, $\forall$, $\wedge$, $\vee$), set operators ($\cap$, $\cup$, $\in$, $\subset$), and operations on the pendant model (*at_instance* (shortcut: @), *temp_select* (shortcut: π_t)).

The eight topological predicates describe the relationship between two regions, and they form the basis of our spatio-temporal uncertain predicates. $inside(A, B)$ means that region A locates inside of region B in geometry. For simplicity, we use notation @ to represent *at_instance*, and π_t to represent *temp_select*. We use the logic operators and set operators to connect terms and form the expressions. We first exam the topological relationship between two moving points. We name the relationship between them at a time instance as an *instant predicate*, and the relationships which lasts for a period as a *moving predicate*. The topological relationship between two static points is either $disjoint$ or *meet*. However, for two moving objects, at a time instant there are three possible relationships. They can be disjoint, or certainly meet, or *possibly meet*. Thus we are able to define the following three instant predicates for moving points with uncertainty.

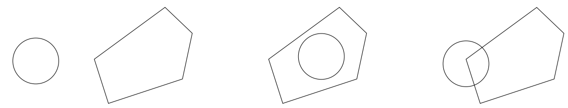
Definition 5 *Assume that we have two moving points* $p, q \in unmpoint$ *and a time instant* t. *Three instant predicates at* t *are defined as follows,*

$$\begin{array}{lll} disjoint\_at(p,q,t) & := & t \in (D(p) \cap D(q)) \ \wedge \ disjoint(@(p,t), @(q,t)) \\ definitely\_meet\_at(p,q,t) & := & t \in (D(p) \cap D(q)) \ \wedge \ @(p,t) \in point \\ & & \wedge \ dist(@(p,t), @(q,t)) = 0 \\ possibly\_meet\_at(p,q,t) & := & \neg disjoint\_at(p,q,t) \ \wedge \\ & & \neg definitely\_meet\_at(p,q,t) \end{array}$$

In the above definition, two moving points are disjoint with each other at t, if the resulting regions from *at_instance* operations are disjoint. If the *at_instance* operation on two objects result two points that are of the same position, they will definitely meet at this time instance. The rest situations all belong to the *possibly_meet_at* relationship. Figure 3 illustrates the above three predicates.

After introducing the instant predicates between two moving points, now we define their moving predicates which describe topological relationships that last for a period of time. Assume that we have an interval $I \subset time$,

Definition 6 $definitely\_encounter(p, q, I) = true$*, if the following conditions hold*



(a) (b) (c)

Fig. 4. Predicates between a moving point and a moving region disjoint_at (a), definitely_inside_at (b) and possibly_inside_at (c)

(i) $I \subset (D(p) \cap D(q)), p_I := \pi_t(p, I), q_I := \pi_t(q, I)$,
(ii) $\exists t_1, t_2, t_3 \in I \ \wedge \ t_1 < t_2 < t_3 \ \wedge \ disjoint\_at(p_I, q_I, t_1)$
$\wedge \ definitely\_meet\_at(p_I, q_I, t_2) \ \wedge \ disjoint\_at(p_I, q_I, t_3)$

Definition 7 *possibly_encounter$(p, q, I) = true$, if the following conditions hold*

(i) $I \subset (D(p) \cap D(q)), p_I := \pi_t(p, I), q_I := \pi_t(q, I)$,
(ii) $\exists t_1, t_2, t_3 \in I \ \wedge \ t_1 < t_2 < t_3 \ \wedge \ disjoint\_at(p_I, q_I, t_1)$
$\wedge \ possibly\_meet\_at(p_I, q_I, t_2) \ \wedge \ disjoint\_at(p_I, q_I, t_3)$

The *definitely_encounter* predicate describes the situation that two moving points will meet for sure during a time interval I. Similarly, *possibly_encounter* means that two moving points will meet with some possibility. Figure 5(a) illustrates this predicate.

Besides the uncertain relationship between two moving points we have introduced above, topological relationships between a moving point and a static region on the land are also of great importance. It can help, for example, detect whether an airplane has the possibility of entering a city when it disappears on the radar. Now, we define some important spatio-temporal uncertain predicates which represent topological relationships between a moving point and a static region on the land.

Definition 8 *Given the movement of a moving point $p \in unmpoint$, a static region $R \in region$, and a time instant $t \in time$, three instant predicates between them are defined as follows,*

$$
\begin{aligned}
disjoint\_at(p, R, t) &:= t \in D(p) \ \wedge \ disjoint(@(p, t), R) \\
definitely\_inside\_at(p, R, t) &:= t \in D(p) \ \wedge \ inside(@(p, t), R) \\
possibly\_inside\_at(p, R, t) &:= \neg disjoint\_at(p, R, t) \wedge \\
&\quad\ definitely\_inside\_at(p, R, t)
\end{aligned}
$$

The above definition formalize three instant predicates between a moving point and a static region. Figure 4(a)-(c) show the above three predicates, where the circle represents the uncertain region of a moving point at the given time instance and the polygon represents the static region. Based on the instant predicates we have defined, we give the following definitions of spatio-temporal uncertain predicates which represent the development of relationships between a moving point and a static region within a period.

Definition 9 *$definitely\_enter(p, R, I) = true$, if the following conditions hold*

(i) $I \subset D(p), p_I := \pi_t(p, I)$
(ii) $\exists t_1, t_2 \in I \ \wedge \ t_1 < t_2 \ \wedge \ disjoint\_at(p_I, R, t_1) \ \wedge$
$definitely\_inside\_at(p_I, R, t_2)$

Definition 10 *$possibly\_enter(p, R, I) = true$, if the following conditions hold*

(i) $I \subset D(p), \ p_I := \pi_t(p, I)$
(ii) $\exists t_1, t_2 \in I \ \wedge \ t_1 < t_2 \ \wedge \ disjoint\_at(p_I, R, t_1) \ \wedge$
$possibly\_inside\_at(p_I, R, t_2)$

Definition 11 *$definitely\_cross(p, R, I) = true$, if the following conditions hold*

(i) $I \subset D(p), p_I := \pi_t(p, I)$
(ii) $\exists t_1, t_2, t_3 \in I \ \wedge \ t_1 < t_2 < t_3 \ \wedge \ disjoint\_at(p_I, R, t_1) \ \wedge$
$definitely\_inside\_at(p_I, R, t_2) \ \wedge \ disjoint\_at(p_I, R, t_3)$

Definition 12 *$possibly\_cross(p, R, I) = true$, if the following conditions hold*

(i) $I \subset D(p), p_I := \pi_t(p, I)$
(ii) $\exists t_1, t_2, t_3 \in I \ \wedge \ t_1 < t_2 < t_3 \ \wedge \ disjoint\_at(p_I, R, t_1) \ \wedge$
$possibly\_inside\_at(p_I, R, t_2) \ \wedge \ disjoint\_at(p_I, R, t_3)$

Definitions 9 to 12 describe spatio-temporal uncertainty predicates between a moving point and a static region. Figure 5(b) illustrates the *possibly_cross* predicate. Similarly, we can define more predicates between an *unmpoint* object and an *unmregion* object, for example, *possibly_enter*, and *possibly_cross*. There are no *definite_enter* or *definitely_cross* relationships between an *unmpoint* object and an *unmregion* object, since there is no *definitely_inside_at* instant predicate between a moving point and a moving region with uncertainty at a time instance. Because of the page limitation, we do not give formal definitions of STUPs between an *unmpoint* and *unmregion* here.

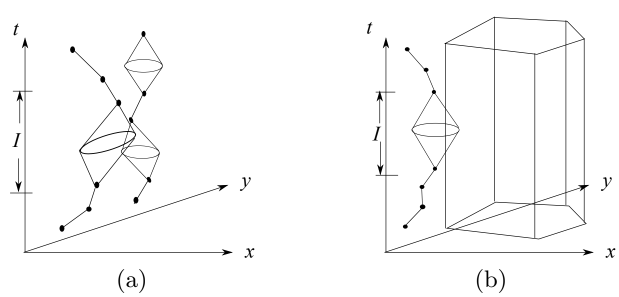


Fig. 5. Spatio-temporal predicates of possibly_encounter (a); possibly_cross (b)

4.2 Spatio-Temporal Uncertainty Queries

Now we discuss how to use STUPs we have defined in Section 4 in database queries. Current database query languages such as SQL are not able to answer temporal queries because they do not support temporal operators. This problem could be solved by implementing the STUPs as operators in queries. Thus, we are able to extend the SQL language to a more comprehensive query language, named as *spatio-temporal uncertainty language* (*STUL*). The STUL language extends SQL and supports the spatio-temporal uncertainty operations in terms of STUPs. Here we show some examples of using this language.

Assume that we have the following database schema of persons,

```
persons(id:integer, name:string, trajectory:unmpoint)
```

The query "Find all persons that may possibly become the witness of the criminal Trudy during the period from 10 am to 12 pm" can be answered by the STUL query as follows,

```
SELECT  p1.id           FROM    persons p1, persons p2
WHERE   possibly_encounter(p1.trajectory,p2.trajectory,
        10:00:00,12:00:00)      AND   p2.name='Trudy'
```

Now we give an example of SQL like query on the predicates between an uncertain moving point and a static region. Assume that we have the following schemas,

```
airplanes(id: string, flight: unmpoint)
airports(name: string, area: region)
```

The query "Find all planes that have possibly entered the Los Angeles airport from 2:00pm to 2:30pm" can be written as follows,

```
SELECT  airplanes.id        FROM    airplanes, airports
WHERE   possibly_enter(airplanes.flight, airports.area,
        14:00:00, 14:30:00) AND airports.name='LAX';
```

We are also able to query on the topological relationship between an uncertain moving point and an uncertain moving region. An example of a moving region with uncertainty is the hurricane. Assume that we have the following schema,

```
hurricanes(name: string, extent: unmregion)
```

The query "Find all planes that have possibly entered the extent of hurricane Katrina between Aug 24 to Aug 25, 2005" can be written as follows,

```
SELECT  a.id            FROM    airplanes a, hurricanes h
WHERE   possibly_enter(a.flight,h.extent,2005-08-24,2005-08-25)
        AND  h.name='Katrina'
```

5 Algorithms to Determine STUP

In this section, we give algorithms to determine the spatio-temporal uncertainty predicates. Because of the page limitation, we take the predicate between two moving points, *possibly_encounter* for example. Algorithms for other predicates will be provided in our future work. In Definition 3, we have stated that each movement is represented by a set of partial functions on a union of intervals, and each interval has its own moving pattern, i.e., either a known movement represented by a linearly function or an unknown movement represented by a double-cone volume. Thus, we are able to fragment the entire movement as a set of slices. The slice model is a discrete approach for representing moving objects, introduced in [4]. A slice is the smallest unit of evaluating the spatio-temporal uncertainty predicate, which is either a line segment or a double-cone volume. The slice unit representation of moving objects with uncertainty is illustrated in Figure 6, where moving object A is represented by $< s_{1,1}, s_{1,2}, \ldots, s_{1,5} >$ with elements ordered by time, and B is represented by $< s_{2,1}, s_{2,2}, \ldots, s_{2,4} >$. We evaluate the predicate between two entire moving objects by evaluating whether an intersection exists between a pair of slices. There are three situations: 1. Both slices are line segments; 2. one slice is a line segment while the other is a double-cone; 3. both slices are double-cone volumes. The first situation is the simplest one in that we can represent two line segments by equations in the 3D plane and compute whether they intersect at a common point, denoted by $comPoint(seg_1, seg_2)$. For situation 2 and 3, since it is cumbersome to calculate the intersection in 3D volumes, we introduce the method to test the intersection only in some time instants, which are called *critical instants*. A line segment has two critical instants: the starting time and ending time respectively. A straight double-cone volume has 3 critical instants: the instants at the bottom apex, the base and the top apex. An oblique double-cone volume has 4 critical instants: the bottom apex, the lower base point, the upper base point and the top apex. Figure 6(b) illustrates critical instants on the three types of volumes. We design an algorithm to compute whether two slices could intersect by testing whether they intersect at critical instants, shown by Figure 7. Because we only exam the intersection at a few number of critical instants, the complexity of this algorithm is constant.

The *possibly_encounter* predicate can then be determined by examining the intersection between pairs of slices. The algorithm is shown in Figure 8. Assume that the first moving object has m slices and the second has n slices, this algorithm will exam $m + n$ times of $unitIntersection$. Since the complexity of $unitIntersection$ is $O(1)$, the total complexity to determine *possibly_encounter* is $O(m + n)$.

6 Conclusions and Future Work

In this paper, we discuss the problem of querying topological relationships between moving objects with uncertainty. We propose the *pendant model* which

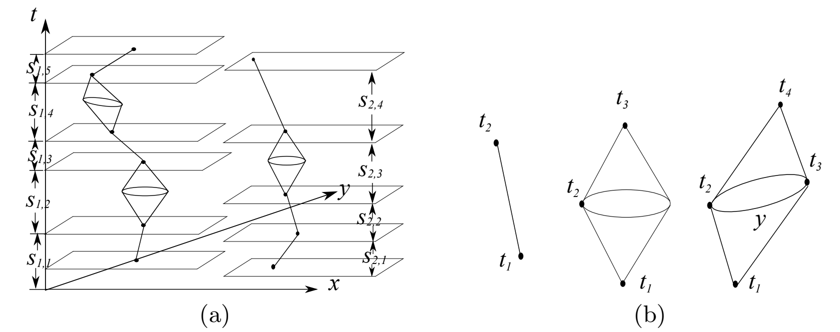


Fig. 6. Slice unit representation of moving objects with uncertainty (a), critical instants of a slice unit with different volumes

```
algorithm unitIntersect (slice S1, slice S2)
  intersect ← false
  S ← empty // sequence of instants
  m ← num_of_critical_instants(S1)
  n ← num_of_critical_instants(S2)
  while {i ≤ m and j ≤ n}
    if time[i] < time[j]
      add time[i] to S; i++
    else
      add time[j] to S; j++
    endif
  endwhile
  add remaining instants of S1 to S
  add remaining instants of S2 to S
  s ← get_first_elem(S)
  while not end_of(S)
        and intersect = false
    u ← at_instance(S1, s)
    v ← at_instance(S2, s)
    if a ∈ point and b ∈ point
      intersect ← dist(u, v) = 0
    endif
    if a ∈ point and b ∈ region
      intersect ← inside(u, v)
    endif
    if a ∈ region and b ∈ region
      intersect ← overlap(u, v)
    endif
    s ← get_next_elem(S)
  endwhile
  return intersect
end
```

Fig. 7. The algorithm *testUnitIntersection* to determines whether two slices intersect

properly represents the spatio-temporal uncertainty of moving objects. We define a new set of predicates called spatio-temporal uncertainty predicates (STUP) which represent moving topological relationships with uncertainty as binary predicates. The benefit of this approach is that the STUPs can be used in queries as selection conditions. We define important STUPs which describe topological relationships between two moving points, or a moving point and a static region, or a moving point and a moving region. We show query examples of how to use them and give an efficient algorithm to compute one of the most important STUPs. In our future work, we will give the algorithms for all other STUPs we have defined. We will study the topological relationships between complex regions with uncertainty as well.

```
algorithm possibly_incounter (umpoint A, umpoint B, t1, t2)
intersect ← false
A′ ← temp_select(A, t1, t2)
B′ ← temp_select(B, t1, t2)
sa ← get_first_elem(A′)
sb ← get_first_elem(B′)
while not (end_of(A′) or end_of(B′))
        and intersect = false
   if sa ∈ segment and sb ∈ segment
      intersect ← comPoints(sa, sb)
   else
      intersect ← unitIntersect(sa, sb)
   endif
   if endtime_of_sa < endtime_of_sb
      sa = get_next_elem(A′)
   else
      sb = get_next_elem(B′)
   endif
endwhile
return intersect
end
```

Fig. 8. The algorithm of *possibly_encounter*

References

1. Erwig, M., Schneider, M.: Spatio-temporal Predicates. IEEE Trans. on Knowledge and Data Engineering (TKDE) **14**(4) (2002) 881–901
2. Trajcevski, G., Wolfson, O., Zhang, F., Chamberlain, S.: The geometry of uncertainty in moving objects databases. In: 8th International Conference on Extending Database Technology (EDBT). (2002) 233–250
3. Hägerstrand, T.: What about people in regional science. Papers in Regional Science **24** (1970) 6–21
4. Forlizzi, L., Güting, R.H., Nardelli, E., Schneider, M.: A data model and data structures for moving objects databases. In: SIGMOD '00: Proceedings of the 2000 ACM SIGMOD international conference on Management of data. (2000) 319–330
5. Güting, R.H., Böhlen, M.H., Erwig, M., Lorentzos, C.S.J.N.A., Schneider, M., Vazirgiannis, M.: A Foundation for Representing and Querying Moving Objects. ACM Trans. on Database Systems (TODS) **25**(1) (2000) 1–42
6. Su, J., Xu, H., Ibarra, O.H.: Moving Objects: Logical Relationships and Queries. In: 7th Int. Symp. on Spatial Temporal Databases (SSTD). (2001) 3–19
7. Trajcevski, G., Wolfson, O., Hinrichs, K., Chamberlain, S.: Managing Uncertainty in Moving Objects Databases. ACM Trans. on Database Systems (TODS) **29** (2004) 463–507
8. Pfoser, D., Jensen, C.S.: Capturing the uncertainty of moving-object representations. In: 6th SSD International Symp. on Advances in Spatial Databases. (1999)
9. Egenhofer, M.J.: Approximations of geospatial lifelines. In: SpadaGIS, Workshop on Spatial Data and Geographic Information Systsems. (2003)
10. Miller, H.J.: A measurement theory for time geography. Geographical Analysis **37** (2005) 17–45
11. Kuijpers, B., Othman, W.: Trajectory databases: Data models, uncertainty and complete query languages. J. Comput. Syst. Sci. **76**(7) (2010) 538–560
12. Hornsby, K., Egenhofer, M.J.: Modeling moving objects over multiple granularities. Annals of Mathematics and Artificial Intelligence **36**(1-2) (2002) 177–194
13. Rafael Grimson, Bart Kuijpers, W.O.: An analytic solution to the alibi query in the space-time prisms model for moving object data. International Journal of Geographical Information Science (2009)
14. Trajcevski, G., Choudhary, A., Wolfson, O., Ye, L., Li, G.: Uncertain range queries for necklaces. In: Proceedings of the 2010 Eleventh International Conference on Mobile Data Management. MDM '10 (2010) 199–208