Balloon: Representing and Querying the Near Future Movement of Predictive Moving Objects

Hechen Liu & Markus Schneider*
Department of Computer & Information Science & Engineering
University of Florida
Gainesville, FL 32611, USA
{heliu, mschneid}@cise.ufl.edu

ABSTRACT

The spatio-temporal uncertainty is an inherent feature of moving objects. One scenario where uncertainty exists is the movement of moving objects in the future, which results from lacking the knowledge of the prediction method. To solve this problem is useful, for example, to predict the locations of a hurricane and its relationships with points of interest on the land. The solution calls for a sound model to describe and handle the uncertainty properly. This paper introduces such a model which represents the spatio-temporal uncertainty in the near future. We introduce an uncertainty model called the *balloon model* specifically for *future* movements. We discuss how to implement the balloon model in the moving object database and define some important operations on querying the uncertainty.

Categories and Subject Descriptors

H.2.8 [**Database Management**]: Database Applications—*Spatial databases and GIS*

General Terms

Design, Algorithms

Keywords

Moving objects, uncertainty, balloon

1. INTRODUCTION

Moving objects such as hurricanes, cars and animals have been studied intensively in the past decade in terms of moving object database [9, 8]. The movement of a moving object can be tracked by some GPS devices which report its location periodically. However, in the periods when a moving object is not recorded, its location is not deterministic, and the uncertainty exists. One scenario where the uncertainty exists is the movement of moving objects in the future. We may ask, for example, whether a hurricane will possibly enter a specific region on the land during a period of time. Since hurricanes can move freely in the 2D Euclidean space and are not limited by constraints such as road networks, their movements in the future is unknown. However, if we obtain the knowledge related to the pattern of a hurricane's movement, such as its movement function at the current moment, the location of this hurricane at a time instance in the near future could be predictable. Queries like "What is the probability that Hurricane Katrina will traverse Florida in the next 5 days?" could be answered.

The first goal of the paper is to represent the spatio-temporal uncertainty of moving objects in the near future. We introduce a spatio-temporal uncertainty model, called the *balloon model*, which was first proposed by one of the authors in [15]. In that paper, we introduce some basic concepts regarding the representation of the historical and future movements. However, the problems such as how to implement the model in databases, as well as some important operations and their algorithms, have not been addressed. In this paper, we propose a comprehensive view of the balloon model by discussing the uncertainty model together with the implementation concepts and the algorithms of some functions. The balloon model is applied specifically to handle the uncertainty of moving objects in the near future. Instead of restricting the uncertainty movement of a moving object to a cylinder or a cone, the balloon model does not have a maximum speed constraint and thus can be applied to more movements in general, i.e., it can represent the uncertainty in various kinds of movements. An advantage of this is that one can apply the uncertainty queries without knowing how the uncertainty is formed, since they are provided by the domain experts. For example, a hurricane researcher can provide us the various wind speed and wind density of a hurricane in a period of time, then the balloon model will help them generating the uncertainty volume of the hurricane and help them make prediction.

Considering the storage, retrieval and manipulating aspects in the database context, the handling of queries on uncertain moving object data is important since a large amount of queries will be generated and processed. The second goal of the paper is to provide a solution to query the uncertainty in the near future. We introduce a set of powerful operations

*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.

SSO '11 Chicago, Illinois USA

on the proposed balloon model, which can be further integrated into database query languages. This will enable the use of the database language to query the near future movement. The benefit of this approach is that the operations can be used as selection and join conditions. Users who are comfortable with the database query languages such as SQL will understand it and use it smoothly, and more complex predicates can be defined on the basis of them. These operations can be integrated into an SQL-like query language. Assume that we have two database relations,

```
airplanes  (flightNo:string, flight:balloon)
hurricanes (name:string, extent:balloon)
```

where *balloon* denotes the uncertainty data types. We may want to detect all airplanes that will possibly cross the projected extent of hurricane Katrina, then we can write the following query,

```
SELECT flightNo
FROM   airplanes, hurricanes
WHERE  hurricanes.name="Katrina" AND
       possibly_cross(flight, extent)
```

The contributions of this work are,

- It formalizes the uncertainty of moving objects in the future and their properties as a database model, which will further enable the query step easier.
- It introduces the operations on the model and how to query the spatio-temporal uncertainty on the basis of these operations.

The rest of the paper is organized as follows. Section 2 discusses related work on uncertainty models and queries in spatio-temporal database. Section 3 proposes our balloon model representing the uncertainty of moving objects. Section 4 discusses how to integrate the balloon model in databases, and introduces the operations on the balloon model and provides the algorithms. Section 5 draws conclusions and discusses the future work.

2. RELATED WORK

In this section, we classify the related literature of this work into two categories, the uncertainty models for moving objects and querying moving objects in databases.

A famous uncertainty model is the *3D cylinder model* [17]. The possible location of a moving object at a time instant is within a disc representing the area of uncertainty. Thus, the trajectory of a moving point is not a polyline but a cylinder in the 3D (2D+time) space. The cylinder model assumes that the degree of uncertainty is constant in a period of time. A comparable approach, the *space-time prism* (beads) *model* [14, 5, 13, 16] represents the uncertain movement of a moving object as the union of two half-cones (a bead) in the 3D space. Given the source location and the definition location, as well as the maximum speed of a moving object, all possible trajectories between two locations are bounded within the bead. The beads model reduces the uncertain volume by two thirds, according the geometric properties of cones. Besides the 3D cylinder model and the beads model, there are some other approaches dealing with spatio-temporal uncertainty. [11] proposes a path prediction method for range queries. [18] introduces the uncertain trajectories hierarchy method to solve the problem of uncertain probabilistic range queries. Both methods are applied in a road network environment. [10] designs a hybrid prediction model for querying the position of moving objects in the near future, by considering both the pattern information and the existing motion functions. [2, 3] propose the *uncertain region* concept to solve probabilistic range queries (PRQ) and probabilistic nearest-neighbor queries (PNNQ). Our previous own work in [12] proposes the *pendant model*, which is based on the space-time prism model to represent the spatio-temporal uncertainty. In particular, we propose the spatio-temporal uncertainty predicates (STUP) to model the uncertain topological relationship between moving objects. The authors have proposed a new model called the balloon model of querying the historical and future movement in [15]. In this paper, we will give a comprehensive view of the balloon model by introducing the implementation concept of the balloon data type in databases, the operations on the data as well as their algorithms.

The study on moving objects includes an important aspect: querying moving objects. [4] proposes *Spatial SQL*, which is a minimal extension to the interrogative part of SQL. It preserves the SQL concepts, and treats spatial objects at a high level by incorporating the spatial operations and relationships. Similarly, spatio-temporal query languages are built on top of SQL as well. Before creating spatio-temporal query language, an important step is to represent moving objects as spatio-temporal data types such as moving point, moving lines and moving regions [8]. Based on the data types, a concept called spatio-temporal predicate is proposed which is later integrated into spatio-temporal query language [6]. The evolving relationships between moving objects are represented as binary predicates, such as *cross* and *enter*, which are later integrated into the spatio-temporal query language. Thus users can use the binary predicates in the SQL-like languages easily.

3. BALLOON: REPRESENTING THE UNCERTAINTY OF FUTURE MOVEMENTS

In this section, we describe our uncertainty model for querying moving objects. We first show some observations on the spatio-temporal uncertainty problems and how they are solved using the space-time prism model (beads). Then, we introduce our *balloon model*. This model is a generic one since it incorporates different applications, especially for those free movements in the 2D space such as airplanes, hurricanes and freely driving cars.

3.1 Observations on Spatio-temporal Uncertainty

The study of spatio-temporal uncertainty is to represent all the possible trajectories of a moving object, with a limited number of certain spatio-temporal records (GPS points) given. Assume that a moving object travels from the source location $A(x_0, y_0)$ at time t_0 and heading to the destination $B(x', y')$ at t', then a list of GPS points are recorded as $p_0(x_0, y_0, t_0)$, $p_1(x_1, y_1, t_1)$, $p_2(x_2, y_2, t_2)$, and $p'(x', y', t')$ respectively, with the condition $t_0 < t_1 < t_2 < t'$, as illustrated in Figure 1 (a). The movements at time instances other than when the query points are recorded are uncertain. If the object travels the minimal distance between two

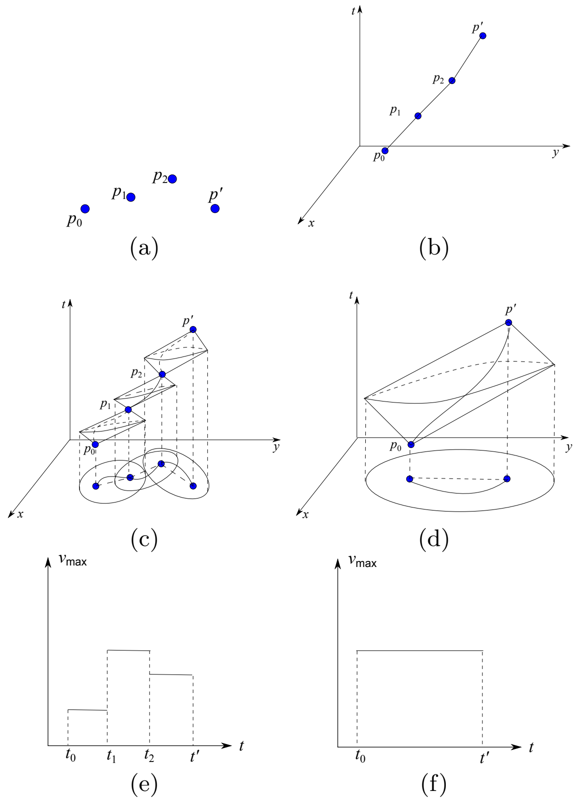


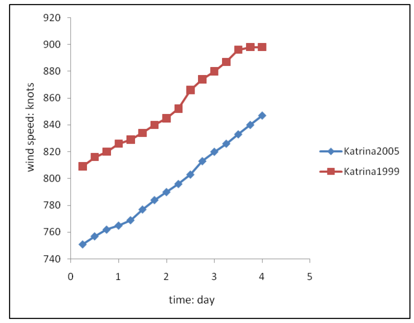
Figure 1: A set of GPS points (a), certain trajectory (b), uncertainty trajectory using the beads model(c) and uncertainty trajectory with lower sampling rate using the beads model(d), the change of the speed in c (e) and the change of the speed in d (f)

consecutive locations, we can approximate the movement as a linear function of time, as illustrated in Figure 1 (b). This can be seen as a certain movement. Considering the uncertainty feature, the movement can be represented using "beads", as illustrated in Figure 1 (c). The reason is that, at any time instance when the position of the moving object is not recorded, its possible position is within an area. If the speed of the moving object cannot exceed a threshold, all the possible trajectories of this moving object between two consecutive query points are within a double-cone volume [5]. The projection of the movement in 2D space is a series of ellipses [14]. The more frequently the locations are updated, the less the uncertainty. If we only record the locations at the source and the destination of the movement, the uncertainty volume is within a larger bead, as illustrated in Figure 1 (d). The speed of the moving point is bounded by the maximum speed between two consecutive locations, as illustrated in Figure 1 (e) and (f) respectively.

Figure 1 (c) shows that the movement between two uncertain query points has a maximum speed as the threshold, thus we can treat the speed to be constant between two consecutive locations. In the real world, we observe that not all movement follow the same pattern as the one in Figure 1 (c). For example, we study the hurricane data of Atlantic Ocean, provided by the National Hurricane Center (NHC) [1]. The sensors keep tracking the hurricane and update the sensed data of the hurricane every 6 hours. The format of

```
60910 10/28/1999 M= 5 11 SNBR=1271 KATRINA   XING=0 SSS=0
60915 10/28*  0   0   0    0*  0   0   0    0*  0   0   0    0*114 809  30 1001*
60920 10/29*116 816  30 1001*120 820  30 1001*126 826  30 1000*132 829  35 1000*
60925 10/30*138 834  35  999*141 840  30 1000*143 845  25 1001*147 852  25 1003*
60930 10/31*160 866  25 1005*172 874  25 1007*184 880  25 1008*194 887  25 1009*
60935 11/01*199 896  20 1010*204 898  20 1011*212 898  20 1011*  0   0   0    0*
```

(a)



(b)

Figure 2: The data of hurricane Katrina captured by sensors in year 1999 (a) and the changing wind speed of Katrina observed in the first 4 days in year 1999 and 2005 respectively

the data file is illustrated in Figure 2 (a), which shows the observed Hurricane Katrina in the first 4 days in year 1999. The wind speeds are encapsulated by the square. From the data we can find that the wind speed of the hurricane varies. Similar situation happens to Katrina in year 2005. If we draw the change of the wind speed in the first four days of Katrina both in year 1999 and 2005 respectively, we can get the diagram shown in Figure 2 (b). We find that the wind speed is not constant but increasing with time. As the speed grows larger, the hurricane travels farther. Therefore, the uncertainty becomes larger and larger in the future as time passing by. The cone is not enough to bound all possible trajectories since it has the maximum speed restriction. Therefore, an alternative model to represent the uncertainty of the movement in the near future is needed. In order to introduce the model properly, we first define some important concepts as follows.

Definition 1 (Historical Movement) *Let point denote the spatial point data type in the 2D space, t_{now} denote the current time, and τ denote the historical movement constructor. The historical movement of a moving point is defined as,*

$$\begin{aligned}
&\tau(point) = \{f : time \rightarrow point \mid \\
&i)\ \ dom(f) = \textstyle\bigcup_{i=1}^{n}[t_{2i-1}, t_{2i}], \forall t \in dom(f) : t \leq t_{now} \\
&ii)\ \forall 1 \leq i < n : t_{2i} < t_{2i+1} \\
&iii) \forall 1 \leq i \leq n : \lim_{t \rightarrow t_{2i-1}^{+}} f(t) = f(t_{2i-1}), \\
&\qquad\qquad \lim_{t \rightarrow t_{2i}^{-}} f(t) = f(t_{2i}), \\
&\qquad\qquad \forall t \in (t_{2i-1}, t_{2i}), f(t)\ is\ derivable\ \}
\end{aligned}$$

In Definition 1, i) and ii) state that the moving point function is defined in a union of non-overlapping intervals of the historical time. iii) states that the function is continuous in its domain.

Definition 2 (Future Movement) *Let region denote the spatial region data type in 2D space, t_f denote a near future*

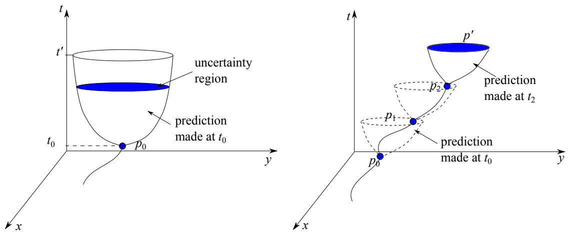


Figure 3: Representing the uncertainty of near future movements using balloon.

time instance,which is the threshold of future time defined by the user, and ψ denote the near future movement constructor. The future movement of a moving point is defined as,

$$\psi(region) = \{f : time \rightarrow region \mid$$
$$i)\ dom(f) = (t_{now}, t_f]$$
$$ii) \forall t \in dom(f), f(t) \text{ is closed}\}$$

In Definition 2, i) shows that the future movement is defined from now to a time threshold in the future. ii) shows that the value of the function at any instance in the future is a closed region.

After obtaining the observations of the historical movement and the future movement of a specific moving object, we are able to introduce the balloon concept.

3.2 Balloon: Representing the Near Future Movement with Uncertainty

As discussed in Section 3.1, there are two parts composing a moving point object, i.e., the historical movement and the future movement. The historical movement can be represented as a polyline in the 2D+time space, and the future movement is an uncertain volume. Therefore, we use the term *balloon* to represent such kind of movement. The historical movement corresponds to the string of the balloon, and the future movement corresponds to the body of the balloon. Figure 3 (a) shows the near future movement of a moving object predicted at the current time t_0. The past movement before t_0 is already known and thus is represented as a function from time to point. The curve of the function is a polyline. The future movement is within the body of the balloon. Since the speed of the moving object varies and is potentially growing with time, we no longer use the cone to represent it. Figure 3 (b) shows different balloons that are predicted at different time instances. Since the position of an observed moving object is updated all the time, the predicted locations varies at different time instances. For example, the predicted uncertain volume containing all the possible trajectories of the near future movement at time t_0 is shown as the lowest balloon in Figure 3 (b). At time t_1, the trajectory of the moving point is updated, and there is a new balloon which represented the movement of the balloon right after t_1. As the speed increases, we see a growing uncertainty over time.

Now, we are able to define our balloon type which contains the past movement and the near future movement together.

Definition 3 (Balloon) *Let Ω denote the constructor of a composed movement function which integrates the movement of a moving point in the past and in the near future, then*

$$\Omega(point, region) = f : \tau(point) \times \psi(region)$$

The above definition introduces our balloon model. We define *balloon* as a new data type to represent moving objects. A balloon data is concatenated by the historical and future movements of a moving object. A special feature is that it takes the uncertainty of the near future movement into consideration. Similar to the cylinder model and the beads model, at any time instance, the possible location of a moving object in the near future is not a single point in the 2D space, but is within an area potentially. Now we give the definition of the concept *uncertainty region*, as shown in Figure 3 (a).

Definition 4 (Uncertainty Region) *Let f denote a balloon movement function, $t \in dom(f)$ is a time instance, $f(t) = S$ is called the uncertainty region at instance t.*

Lemma 1 *The area of the uncertain region for a moving point at any instances in its historical movement is 0.*

Proof. Let t_0 denote the current time, obviously, if $t < t_0$, $f(t) \in point$, then $area(f(t))$=0 □

As we have mentioned above, since the speed of the moving object varies over time, its uncertainty also changes. We formalize a situation in which the degree of uncertainty is increasing, as stated in Definition 5.

Definition 5 (Uncertainty Growing) *Let f denote a balloon movement function, $I \subset dom(f)$. I is called the Uncertainty Growing Interval, if*

$$\forall t_1, t_2 \in I, t_1 < t_2 : area(f(t_1)) \leq area(f(t_2))$$

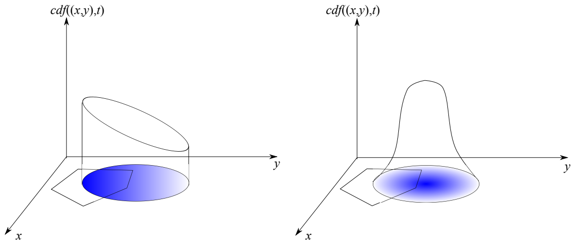
Now, we give the definition of *confidence*, which shows the degree of certainty that a spatial 2D object(point or region) at a time instance is the potential location of a moving object.

Definition 6 (Confidence) *Let f denote a future movement function, $t \in dom(f)$, a spatial object $so \in \alpha$ and $\alpha \in \{point, region\}$, then $con(so, t)$ is a value between $[0, 1]$ that shows the confidence of the 2D spatial object so to be traveled by the moving point at instance t. It satisfies the following conditions,*

$$i)\ \forall t > t_{now}, so \in point : con(so,t) = 0$$
$$ii)\ \forall t > t_{now}, so \in region : con(so,t) \neq 0$$
$$iii) \forall t \leq t_{now}, f(t) = p : conf(p,t) = 1$$
$$iv) \forall t > t_{now}, f(t) = S : conf(S,t) = 1$$

Lemma 2 *The confidence of a spatial point p at any instance of a historical movement is either 0 or 1.*

Proof. In the historical movement, when $t < t_0$, $f(t) \in point$. If $p \in f(t)$, p is the exact location that has been traversed by the moving point, according to Definition 6 (iii). Then if p is not in $f(t)$, its confidence to be the location traversed by the moving point is 0. □



(a) (b)

Figure 4: Uncertain regions with different confidence distribution functions.

Definition 7 (Confidence Distribution Function) *Let f denote a future movement function, $t \in dom(f)$, and r denote a region, $p = (x, y)$ is a spatial point and $p \in r$, then $cdf((x, y), t)$ is a function called confidence distribution function which shows the density of the confidence distributed at different point in a region. It satisfy the following condition,*

$$i) \forall t > t_{now}: \\ con(r,t) = \iint_{p=(x,y)\in r} cdf((x,y),t)dxdy$$

$$ii) \forall t > t_{now}, f(t) = S: \\ con(S,t) = \iint_{p=(x,y)\in S} cdf((x,y),t)dxdy = 1$$

Lemma 3 *If the confidence is uniformly distributed in the uncertainty region $f(t)$ at any instance t, then the confidence of a spatial point p in a future movement is monotonically decreasing with time in an uncertainty growing interval.*

PROOF. Since the sum of the confidence for all points in the uncertainty region of a time instance is 1, in the uncertainty growing interval, the area of the uncertainty region is growing with time. Therefore, if the confidence of the uncertainty region is uniformly distributed inside the region, the confidence of a certain region will be non-decreasing. □

The confidence distribution function shows how the confidence are distributed among the 2D space and varies from different applications. The benefit of introducing this concept is that the confidence distribution function is provided by domain specific experts, thus the users can directly apply it to predict the near future movement. Figure 4 (a) and (b) show two different confidence distribution functions. Figure 4 (a) shows that the *cdf* value of points is decreasing when traversing to the right of the region. Figure 4 (b) shows the case in which the *cdf* is a normal distribution function. If the two uncertain regions overlap with a same polygon in the 2D plane, although the two overlapping parts have the same area, the confidences of the two areas are different.

4. DATABASE MODEL

In this section, we introduce the concept of implementing the balloon model into databases. This includes creating the balloon data type in the databases that can enable users to create instance on it (Section 4.1), as well as a set of functions and predicates which can be used in uncertainty queries (Section 4.2 and Section 4.3).

4.1 Creating the Balloon Data Type in Existing Databases

The balloon data type is defined on the basis of primitive data types that have already existed in the database. The underlying data types include *integer*, *string* (or *varchar*), *timestamp*, as well as spatial data types such as *point*, *region* that have been supported by many nowadays commercial databases.

For the implementation purpose, we define the balloon data type at the discrete level, that we only store the uncertainty region of the balloon at some time instances in the database. We give the definition of the balloon data type for databases in Definition 8.

Definition 8 (Balloon Data Type) *A balloon data object in databases is composed by a finite set of tuples,*

$$< (t_0, t_1, p_1, r_1, c_1), \ldots (t_0, t_i, p_i, r_i, c_i), \ldots >$$

with the following conditions,

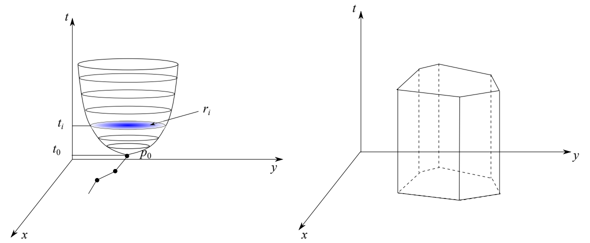
i) $c_i \in [0, 1]$ *or* $c_i = null$
ii) t_0 *is the instance of the current time*
iii) if $t_i \leq t_0$, *then* $r_i = null, c_i = null$
iv) if $t_i > t_0$, *then* $p_i = null$

A tuple $(t_0, t_i, p_i, r_i, c_i)$ is called a "slice".

The above definition shows that the balloon database type consists of a set of tuples and each tuple is composed by 5 attributes, which are two timestamps, one point, one region and one numeric number. In i) c_i denotes the confidence distribution function. ii) states that t_0 is an indicator of the *current time* when the database was updated at the last time. This is important since the database can be updated frequently and the uncertainty may change over time, thus a future movement may become a historical movement with the updating of the database. iii) states that if t_i is less than or equal to the current time, it is the historical part of the movement, then the movement at this instance is a point object, thus uncertainty region and the confidence distribution function have null values. iv) states that if t_i is greater than the current time, it is the future movement and thus the uncertainty is represented as a region with a confidence distribution function, therefore, the point value is null.

Definition 8 gives a discrete representation of the balloon data type in databases and enables the approximation of the uncertainty volume. It is similar to our approach of approximating smooth movements of moving objects in [7, 12]. Under this approach, a balloon object can be represented as shown in Figure 5 (a). To represent the historical movement, we store the trajectory of the moving object which is composed by line segments. To represent the future movement, we store the predicted snapshots of the uncertainty moving object at future time instances. We call a historical point or a future uncertainty region as a *slice*. The transition of the uncertainty region between consecutive time instances is constructed through linear approximation. Our slice approach can also represent a 2D region object in terms of a spatio-temporal object, with the assumption that the position and the extent of the region keep constant over time.

Definition 9 (Uncertainty Relation) *An uncertainty relation of moving objects has the following attributes,*
moving objects (indicator:string, movement:balloon)



(a) (b)

Figure 5: Slice representation of a balloon data (a) and a region object (b).

An example of the uncertainty relation is similar to the one we have mentioned in Section 1:

```
airplanes  (flightNo:string, flight:balloon)
```

where the *flightNo* attribute is an identifier of an airplane object, and the flight attribute denotes the uncertain movements of the airplane.

4.2 Instant Operations on Balloon Data

In the following two subsections, we define some important operations on the balloon data type, which are implemented as user defined functions of the database.

In [12], we have defined the *at_instance* operation. Similarly, we define two corresponding operations on the balloon type, which are *at_past_instance* and $at\_future\_instance$, as shown in Definition 10 and Definition 11 respectively.

Definition 10 (at_past_instance) *Given a balloon object representing the uncertainty movement of a moving point, and a time instance before or at the current time, the at_past_instance operation will return the location of the moving object at that past instance as a point. It has the following signature,*

$$at\_past\_instance : balloon \times instant \rightarrow point$$

The algorithm of *at_past_instance* is shown in Figure 6. *get_next_slice* is the operation of finding the next slice of the current slice. In the algorithm, we first find the interval that containing the time instance, which can be achieved in $O(\log n)$ by performing binary search on t, where n is the number of slices in the balloon object. Then we find the point from the specific interval by calling *get_instance_in_slice* method, as shown in Figure 7. The *get_instance_in_slice* takes $O(1)$ time, thus the total complexity is $O(\log n)$.

The operation *at_future_instance* is defined similarly. The $get\_future\_instance(f, s1, s2, t))$ will find the region at an instance between two stored slices. The uncertainty movement between two time slices is part of the balloon, which is a 3D volume. This volume is represented as a function of time by approximation, thus, we can approximate the value of the function at time t. The algorithms of these two functions are shown in Figure 8 and Figure 9 respectively. The $at\_future\_instance$ also takes $O(\log n)$ time.

Definition 11 (at_future_instance) *Given a balloon object, and a time instance after the current time, the at_future_instance operation will return the uncertainty region of the moving object at that future instance. It has the following signature,*

```
input A balloon object bo, a time instance t
output A point p
method at_past_instance(bo, t)
  if t > now
    return null
  else
    s1 := the last slice in bo that s1.t <= t
    if s1.t = t
      return s1.p
    else
      s2 := get_next_slice(s1)
    return (get_past_instance(s1, s2, t))
  end
```

Figure 6: The at_past_instance algorithm

```
input two point slices s1, s2, a time instance t
output A point p
method get_past_instance(s1, s2, t)
  if t < s1.t or t > s2.t
    return null
  else
    f := linear function determined by
          (s1.t, s1.p), (s2.t, s2, p)
    p := f(t)
    return p
  end
```

Figure 7: The get_past_instance algorithm

$$at\_future\_instance : balloon \times instant \rightarrow region$$

```
input A balloon object bo, a time instance t
output A region r
method at_future_instance(bo, t)
  if t <= now
    return null
  else
    s1 := the last slice in bo that s1.t <= t
    if s1.t = t
      return s1.r
    else
      s2 := get_next_slice(s1)
    return (get_future_instance(f, s1, s2, t))
  end
```

Figure 8: The at_future_instance algorithm

The $confidence\_at$ operation is also of significant meaning. It describes how certain (or uncertain) a region can intersect with a moving object in the future. We give the following definition,

Definition 12 (confidence_at) *Given a balloon object, a region in the 2D space, and a time instance in the future, the confidence_at operation will return the confidence of the moving point to be potentially inside the region at that time instance. It has the following signature,*

$$confidence\_at : balloon \times region \times instance \rightarrow real$$

The algorithm of $confidence\_at$ is shown in Figure 10. The *Intersection* operation is a set operation, which will

```
input two region slices s1, s2, a time instance t
output A region r
method get_future_instance(f, s1, s2, t)
if t < s1.t or t > s2.t
    return null
else
    v := volume function determined by
            (s1.t, s1.r), (s2.t, s2.r)
    r := v(t)
    return r
end
```

Figure 9: The get_future_instance algorithm

```
input A balloon object bo, a region r,
    a time instance t
output A decimal number d between [0,1]
method confidence_at(bo, r, t)
d := 0
if t > now
    r1 := at_future_instance(bo, t)
    r' := Intersection(r, r1)
    d := Integral all (x, y) ∈ r' with bo.cdf
return d
end
```

Figure 10: The confidence_at algorithm

return the common part of two regions. This operation will call the $at\_future\_instance$, thus it calls $O(\log n)$ time.

4.3 Periodical Operations on Balloon Data

In comparison with the instant operations on the balloon data we introduced above, we define some operations which last for a period of time. We first give an important operation *temporal_selection*, as shown in Figure 11.

Definition 13 (temporal_selection) *Given a balloon object, and a time interval $I = [t_1, t_2]$ in the balloon's life time, the temporal_selection operation will return part of the balloon object. It has the following signature,*

$$temporal\_selection : balloon \times interval \rightarrow balloon$$

The *temporal_selection* will find all slices between the interval. The first slice in the interval may have t greater than the start of the interval, then we will add the previous slice to the result. Similarly, the time of the last slice in the interval might be less than the end point of the interval, and we add the next slice to the result as well. It will traverse all slices in the worst case, ths the complexity is $O(n)$.

Now, we define the operation *had_crossed*. This operation is an boolean predicate on the historical part of the movement that tells whether a moving point had crossed a specific region in the given time interval. "cross" indicates a transition of different relationships between the moving point and the region. The moving point is disjoint with the region at first, then inside the region, then disjoint the region again, as illustrated in Figure 14 (a). The algorithm of *had_crossed* is shown in Figure 12.

From the above subsection, we have defined how to calculate the confidence of a 2D region to be traversing by an

```
input A balloon object bo, an interval I,
output A balloon object rbo
method temporal_select(bo, I)
rbo := null
if t1 < bo.mintime or t2 > bo.mintime
    return null
else
    s1 := the first slice that s1.t > I.t1
    s11 := get_prev_slice(s1)
    s2 := the last slice that s2.t < I.t2
    s22 := get_next_slice(s2)
    s := s11
    while s not s22
        rbo := rbo ∪ s
        s := get_next_slice(s)
    add all slices between s11 and s22 to rbo
return rbo
end
```

Figure 11: The temporal_selection algorithm

```
input A balloon object bo, a region r, an interval I,
output A boolean value
method had_crossed(bo,r,I)
if I.t2 >= now
    return false
else
    bo' := temporal_selection(bo, I)
    sa := the first slice in bo'
    sb := the last slice in bo'
    while disjoint(sa.p, r)
        sa := get_next_slice(sa)
    while disjoint(sb.p, r)
        sb := get_prev_slice(sb)
    if sa.t <= sb.t and inside(sa.p, r)
        return true
    else
        return false
end
```

Figure 12: The had_crossed algorithm

uncertain moving point. Since the confidence value is between [0,1], we partition the interval into four sub-intervals. Then, we introduce four terms that indicates the different degrees of confidence.

possibly indicates that the confidence is in greater than 0.25. *likely* means that the confidence is in greater than 0.5. *probably* means that the value is greater than 0.75, and definitely means that the confidence value is 1. Therefore, we can use these terms to show different degree of uncertainty. For example, the predicate *probably_enter* will tell whether a moving object will enter a 2D region with a confidence greater than 0.75. Now, we define the *possibly_enter* operation under the balloon model. Its algorithm is shown in Figure 13.

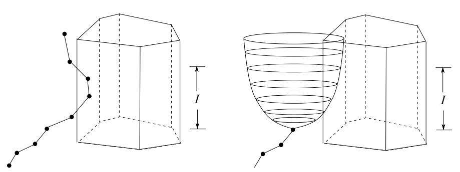
Definition 14 (possibly_enter) *Given a balloon object, a region in 2D space, and a time interval, the possibly_enter operation will return a boolean value, indicating the maximum confidence of the object entering the region in the future is no less than 0.25. It has the signature,*

```
    input A balloon object bo, a region r, an interval I,
    output A boolean value
    method possibly_enter(bo,r,I)
1   max_conf := 0
2   bo' = temporal_select(bo, I)
3   if I.t1 < now or confidence_at(bo, r, I.t1) > 0
4     or confidence_at(bo, r, I.t2) = 0
5     return false
6   else
7     for each slice s in bo'
8       conf := confidence_at(s.r, bo', s.t)
9       if conf > max_conf
10        max_conf = conf
11    if max_conf >= 0.25
12      return true
13    else
14      return false
15  end
```

Figure 13: The possibly_enter algorithm

$$possibly\_enter : balloon \times region \times interval \rightarrow bool$$

Line 2 performs a temporal selection given the time interval. $I.t1$ and $I.t2$ denote the start and end time instance of I respectively. Line 3 to Line 4 indicate that the moving object should be disjoint with the region at $I.t1$ and should have intersection with the region at $I.t2$. Line 7 to Line 10 calculate the confidence of the region in each slice, and keep recording the maximum one. If the maximum confidence is greater than or equal to 0.25, the function will return *true*.



(a) (b)

Figure 14: The *had_cross* predicate(a) and *possibly_enter* predicate (b).

5. CONCLUSIONS AND FUTURE WORK

In this paper, we address the problem of modeling and querying the uncertainty of a moving object in the near future in the database context. We introduce the balloon model, which represents the uncertainty of moving objects in the near future properly. A balloon object is composed by two parts, the body that represents the near future uncertainty, and the string that represents the past certain trajectory. We also propose an implementation concept of representing the balloon model in the database by slice representation. We design the functions of different kinds of uncertainty queries as part of the model and show the algorithms. Our next step will be implementing the system of the balloon model and perform the efficiency experiments on various kinds of spatio-temporal uncertainty queries on the near future movements of moving objects.

6. REFERENCES

[1] National hurricane center archive of hurricane seasons. http://www.nhc.noaa.gov/pastall.shtml.

[2] R. Cheng. Evaluating probabilistic queries over imprecise data. In *SIGMOD*, pages 551–562, 2003.

[3] R. Cheng, D. Kalashnikov, and S. Prabhakar. Querying imprecise data in moving object environments. *IEEE Transactions on Knowledge and Data Engineering (TKDE)*, 16(9):1112 – 1127, 2004.

[4] M. J. Egenhofer. Spatial sql: A query and presentation language. *IEEE Transactions on Knowledge and Data Engineering (TKDE)*, 6:86–95, 1994.

[5] M. J. Egenhofer. Approximations of geospatial lifelines. In *SpadaGIS, Workshop on Spatial Data and Geographic Information Systsems*, 2003.

[6] M. Erwig and M. Schneider. Spatio-temporal Predicates. *IEEE Trans. on Knowledge and Data Engineering (TKDE)*, 14(4):881–901, 2002.

[7] L. Forlizzi, R. H. Güting, E. Nardelli, and M. Schneider. A data model and data structures for moving objects databases. In *SIGMOD*, pages 319–330, 2000.

[8] R. H. Güting, M. H. Böhlen, M. Erwig, C. S. J. N. A. Lorentzos, M. Schneider, and M. Vazirgiannis. A Foundation for Representing and Querying Moving Objects. *ACM Trans. on Database Systems (TODS)*, 25(1):1–42, 2000.

[9] R. H. Güting and M. Schneider. *Moving Objects Databases*. Morgan Kaufmann Publishers, 2005.

[10] H. Jeung, Q. Liu, H. T. Shen, and X. Zhou. A hybrid prediction model for moving objects. In *ICDE*, pages 70–79, 2008.

[11] H. Jeung, M. L. Yiu, X. Zhou, and C. S. Jensen. Path prediction and predictive range querying in road network databases. *The VLDB Journal*, 19:585–602, 2010.

[12] H. Liu and M. Schneider. Querying moving objects with uncertainty in spatio-temporal databases. In *DASFAA*, pages 357–371, 2011.

[13] H. J. Miller. A measurement theory for time geography. *Geographical Analysis*, 37:17–45, 2005.

[14] D. Pfoser and C. S. Jensen. Capturing the uncertainty of moving-object representations. In *SSD*, pages 111–132, 1999.

[15] R. Praing and M. Schneider. Modeling historical and future movements of spatio-temporal objects in moving objects databases. In *CIKM*, pages 183–192, 2007.

[16] G. Trajcevski, A. Choudhary, O. Wolfson, L. Ye, and G. Li. Uncertain range queries for necklaces. In *MDM*, pages 199–208, 2010.

[17] G. Trajcevski, O. Wolfson, K. Hinrichs, and S. Chamberlain. Managing Uncertainty in Moving Objects Databases. *ACM Trans. on Database Systems (TODS)*, 29:463–507, 2004.

[18] K. Zheng, G. Trajcevski, X. Zhou, and P. Scheuermann. Probabilistic range queries for uncertain trajectories on road networks. In *EDBT*, pages 283–294, 2011.