

# Modeling Historical and Future Movements of Spatio-Temporal Objects in Moving Objects Databases

Reasey Praing & Markus Schneider<sup>\*</sup>  
Department of Computer & Information Science & Engineering  
University of Florida  
Gainesville, FL 32611, USA  
{rpraing, mschneid}@cise.ufl.edu

## ABSTRACT

Spatio-temporal databases deal with geometries changing over time. In general, geometries do not only change discretely but continuously; hence we are dealing with moving objects. In the past, a few moving object data models and query languages have been proposed. Each of them supports either historical movements or future movements but not both together. Consequently, queries that start in the past and extend into the future cannot be supported. To model both historical and future movements of an object, two separate concepts with different properties are required, and extra attention is necessary to avoid their conflicts. Furthermore, current definitions of moving objects are too general and vague. It is unclear how a moving object is allowed to move through space and time. For instance, the continuity or discontinuity of motion is not specified. In this paper, we propose a new moving object data model called *Balloon model* which provides integrated support for both historical and future movements of moving objects. As part of the model, we provide formal definitions of moving objects with respect to their past and future movements. All kinds of queries including past queries, future queries, and queries that start in the past and end in the future are supported in our model.

## 1. INTRODUCTION

In recent years, with the advances in storage, sensor, and database technologies, considerable progress has been made to manage temporally dependent spatial data. Across many diverse disciplines including robotics, mobile computing, and geographical science, the growing demand for handling changing geometric objects has fueled research in spatial and temporal databases and led to their fusion into spatio-temporal databases. We are especially interested in the design of data models for continuously changing geometric objects over time that are typically known as *moving objects*.

Although there have been some developments with respect to modeling and managing moving objects, we have identified three main problems. The first problem concerns the modeling of historical

<sup>\*</sup>This work was partially supported by the National Science Foundation under grant number NSF-CAREER-IIS-0347574.

movements of moving objects. Existing approaches define a moving object data type as a function from time to space. This concept is very useful since it can be applied to construct a moving data type from any non-moving type. However, the lack of well defined properties of the mapping function makes it very general and vague. It is unclear how a moving object can move or evolve in space and time, and whether the function representing the movements is a total or partial function. In fact, the concept of movement and evolution including their properties such as continuity, splitting, and disappearing, as they are applied to moving objects, have not been investigated. The second problem concerns the modeling of future movements of moving objects. Existing models tend to support the near future movements of moving objects in specific environments, e.g., a constrained network environment, by providing a specific method for predicting and querying the movements. Recently, a new research development has taken a different approach by focusing on defining a generic data model and handling the inherent uncertainty aspect of future movements of moving objects instead of predicting it. In any case, the time to space function is implied such that giving a time instant in the future, the potential positions of the moving object are provided. Again, the lack of a precise specification of the properties of future movements and evolutions makes it unclear how a moving object can move or evolve in the future. Finally, the third problem is the lack of an integrated and unified model which can support both historical and future movements of moving objects. So far, existing models support either historical movements or future movements but not both together. Modeling the past and future movements of an object requires two separate concepts. This requires extra attention to synchronize both concepts in a single model such that they can be applied to the same object without conflicting each other. Another drawback is the difficulty or lack of support for *past-future spanning* queries, i.e., queries that evaluate moving data that start from the past and extend into the future.

The goal of this paper is to solve these problems by introducing a new model called *Balloon model* to support both historical and future movements of moving objects. We use the metaphor of a *balloon* to model a moving object: the *string* and the *body* of the *balloon object* represent the past and the future movement respectively. For example, the projected past movement and future prediction of the eye of a hurricane are usually illustrated using a shape that resembles a balloon. The past movement of the eye (a moving point) can be seen over time as a movement along a line or a curve which resembles the string of a balloon. On the other hand, the position of the eye at a time instant in the future can be anywhere within an area of uncertainty. Thus, the future movement of the eye can be seen as a moving region of uncertainty. This moving re-

gion of uncertainty resembles the body of a balloon. Furthermore, the connection point between the string and the body of a balloon object represents the present state of the moving object. With this model, we define a new set of data types for representing different kinds of balloon objects. We solve the first two problems by providing a precise and formal definition of moving objects (balloon objects) along with specifications of the properties of their movements. We solve the third problem by modeling both historical movements and future predictions of moving objects in our model and providing support for new operations and queries.

The paper is presented according to the following structure. In Section 2, we discuss spatial objects as well as existing models for past and future movements of moving objects. In Section 3, we explore how a moving object can move or evolve and define a model for historical development of moving objects. Then, we describe how future movements of moving objects can be represented in Section 4. Based on these definitions, we propose our Balloon model including data type definitions and operations in Section 5. Then, we present different types of queries that can be supported by this model in Section 6. Finally, in Section 7, we draw conclusions and discuss future work.

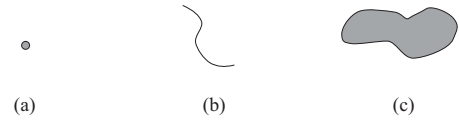
## 2. RELATED WORK

Within the past decade, there have been several major developments of spatio-temporal data models [12]. Among these developments, moving object models have been of significant interest in both application and research directions [9]. Each moving object model concerns about the movement of spatial objects. In Section 2.1, we briefly discuss the different types of spatial objects. Then in Section 2.2, we discuss an existing moving object model that supports the past evolution of moving objects. We take a look at a number of models that support specific types of near future developments of moving objects in Section 2.3.

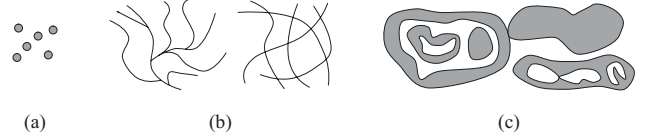
### 2.1 Spatial Objects

The development of data models for spatial data can be distinguished into two generations of models. The first generation offers simple structure objects like single points, continuous lines, and simple regions (Figure 1). Since these simple geometric structures are not able to represent a variety of complex geographic reality, the second generation of data models solves this problem by providing more complex representations of spatial objects that allow support for multi-component objects as well as objects with holes. This development offers complex spatial data types including *complex points*, *complex lines*, and *complex regions* (Figure 2). The formal definition of these data types is defined based on point-set topology and is provided in [14]. Here, we only provide a rough and informal description of these complex spatial types. A complex point object is a finite set of single points. Each of these points is a component of the complex point. A complex line is defined as the union of the images of a finite number of continuous mappings. Each connected set of these images is called a block representing a component of the complex line. A complex region is defined as a set which is *regular closed*, *bounded*, and may contain a finite number of connected sets. Each of these connected sets is called a face, and each face may contain one or more holes. Note that these complex spatial data types allow a special object which is an empty object denoted by  $\perp$  to represent the undefined value that may be a result of spatial operations.

To study the movements and evolutions of these objects, it is essential to have a clear understanding of their definition. It is beyond



**Figure 1: Examples of a simple point object (a), a simple line object (b), a simple region object (c).**



**Figure 2: Examples of a complex point object (a), a complex line object (b), and a complex region object (c).**

the scope of this paper to provide details on this aspect. Thus, we assume that this concept of spatial objects is well understood. In this paper, we define movements and evolutions of both simple and complex spatial objects. Thus, our Balloon model supports both generations of spatial objects. Since simple spatial types are just special cases of complex spatial types, throughout this paper, we will address these spatial data types by *point*, *line*, and *region*.

### 2.2 An Existing Model for Historical Movements of Moving Objects

Some approaches have been taken to represent the (past) movement of moving objects [8, 16]. The past development is a historical fact that is modeled as a function from time to space [1, 6]. For an arbitrary data type  $\alpha$ , the corresponding moving data type of  $\alpha$  is a function  $\tau(\alpha)$  that provides the mapping from the temporal domain to  $\alpha$ , i.e.,

$$\tau(\alpha) = \text{time} \rightarrow \alpha$$

By applying the type constructor  $\tau$  to spatial data types such as *point*, *line*, and *region*, the definition of the corresponding moving data types such as moving point (*mpoint*), moving line (*mline*), and moving region (*mregion*) can be obtained respectively [2].

$$\text{mpoint} = \tau(\text{point}) = \text{time} \rightarrow \text{point}$$

$$\text{mline} = \tau(\text{line}) = \text{time} \rightarrow \text{line}$$

$$\text{mregion} = \tau(\text{region}) = \text{time} \rightarrow \text{region}$$

Since only the domain and range of the function is specified, this function can take any shape and form making these data type definitions vague. For example, the definition allows a moving point to move continuously (Figure 3(a)), disappear for periods of time (Figure 3(b)), jump in space at specific time instants (Figure 3(c)), or even appear instantly (Figure 3(d)). Some of these configurations of a moving point are against our intuitive understanding of movement. Furthermore, these definitions tend to assume that we have precise and complete knowledge about the evolution of spatial objects. Hence, only the past evolution can be modeled, and the temporal domain should not exceed the present into the future. In our Balloon model, we employ a similar approach of using a type constructor to construct moving types for representing historical movements of balloon objects. However, we take the extra step of specifying the properties of objects' movements as well as evolutions and describe how an object can move or evolve in the past.

### 2.3 Existing Models for Future Movements of Moving Objects

With regard to the future movements of moving objects, currently existing models are restricted to specific types of motion. The MOST model [15], which is based on the concept of motion vector, is able to represent near future developments of moving objects through an assumption on the objects' inertia that the current motion direction does not change in the near future. Although, this model is able to represent the near future movement of a moving object, the predictive movement is rather limited to a single motion function. The aspect of uncertainty such as the probability of occurrence is not part of the model. Other models are able to capture the uncertainty aspect of the future movements through the use of a trajectory or motion plan with a constant [7] or variable uncertainty threshold [11]. Another approach presented in [10] models and predicts future movements of moving objects in a constrained network environment. However, an aspect which all of these models have in common is that they are designed to handle a specific type of motion only; they offer or assume a specific prediction technique and provide querying capabilities on the predicted data.

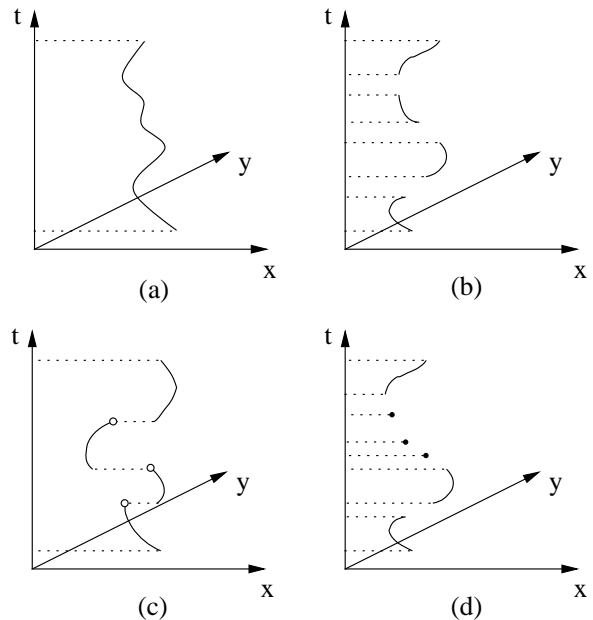
A recent approach called the FuMMO model [13] focuses on modeling how the future movements of moving objects may look like in general instead of providing a specific prediction model for a specific kind of movements. The model emphasizes the separation between the prediction, which should be a task of domain experts, and the generic data modeling of future movements, which a moving object database management system should support. Although this is a considerable step in the right direction, the model uses a similar type construction method as in the case of the historical model and lacks precise definitions of function properties. Furthermore, this model uses spatial data types as the basis for constructing future moving object data types. As a result, the construction procedure is complex due to the number of steps required to specify future moving data types based on spatial data types. In our Balloon model, we take a different approach by specifying the future moving data types for balloon objects based on our historical moving types. In doing so, we reduce the complexity of the type constructions as well as offer a precise definition of the future movements of moving objects.

## 3. MOVING OBJECTS: MODELING HISTORICAL MOVEMENTS

Before we can define a model for moving objects, we must have a clear understanding of the concepts of movement. In this section, we focus on defining how an object can move and/or evolve in space and time. We are especially interested in the movement and evolution of both simple and complex spatial objects. To specify how these objects move or evolve, we first define the properties of movement and evolution including the aspects of continuity, growing, and shrinking for simple spatial objects in Section 3.1. Having defined how a simple object moves or evolves, we can then use these concepts to specify how a complex object moves or evolves and discuss additional problems such as appearing, disappearing, splitting, and merging of object components in Section 3.2. These concepts are then used in Section 3.3 to define moving data types for historical movements of moving objects.

### 3.1 Continuity of Movement and Evolution of Moving Objects

To define the movement or evolution of objects, we first need to define a concept for time and space. Since we are interested in contin-



**Figure 3: Examples of valid movements: a single continuous interval (a) and multiple continuous intervals (b). Examples of invalid movements: instantaneous jumps within an interval (c) and instantaneous appearances (d).**

uously changing information, we define a type  $time = \mathbb{R}$ . In regard to space, we use the two-dimensional Euclidean space. Here, we concern about the developments of two-dimensional spatial objects over time.

Recall that spatial data types are defined to include also the empty object ( $\perp$ ) which is used to represent the nonexistence or undefined value of spatial objects [14]. To model movements for these objects, existing approaches have been to define a moving object as a function of spatial objects over time. For an object of a spatial type  $\alpha$ , a *spatio-temporal mapping* is defined as a function  $\tau(\alpha) = time \rightarrow \alpha$ . Without any constraint or limitation, this function is too general for specifying moving objects since it allows undesirable movements such as instantaneous jumps (Figure 3(c)) and instantaneous appearances (Figure 3(d)). The latter case is obvious, so we only need to define the former case. An instantaneous jump of a spatio-temporal mapping  $f$  at a time instant  $t$  is defined such that  $\lim_{\epsilon \rightarrow 0} f(t - \epsilon) \neq f(t)$ . In Figure 3(c), we indicate an instantaneous jump by a hollow circle and a dashed line connecting the movement before and after the jump. Although these discontinuous movements are impractical in nature, certain types of discontinuity such as periods of unknown movements should be allowed (Figure 3(b)). To accommodate this, we need to specify constraints on the spatio-temporal mapping  $\tau(\alpha)$  such that undesired scenarios are excluded. Thus, we define an *interval-based spatio-temporal mapping* of a type  $\alpha$  as follows:

**Definition 1** A spatio-temporal mapping  $f : \tau(\alpha)$  for a spatial data type  $\alpha$  is an *interval-based spatio-temporal mapping* if there exists a union of intervals  $D \subset time$ ,  $D = [t_1, t_2] \cup [t_3, t_4] \cup \dots \cup [t_{n-1}, t_n]$ ,  $\forall 1 \leq i < n, t_i < t_{i+1}$  such that  $\forall t \in D : f(t) \neq \perp$  and  $\forall t \notin D : f(t) = \perp$ .

While this definition excludes the case with instantaneous appear-

ances as in Figure 3(d), it still allows instantaneous jumps within a defined time interval as in Figure 3(c). To exclude this case, it is necessary to define a concept of continuity for movement or evolution within each *defined interval* (an interval of  $D$ ) of the domain. More precisely, we regard only *interval continuous* spatio-temporal mappings as valid movements or evolutions. Since our spatial objects are defined based on point-set topology [14], each object is a point-set consisting of point *elements*. To formally define continuity, we employ a concept of distance between point elements of spatial objects. We define a distance operation  $\delta : \text{point} \times \text{point} \rightarrow \mathbb{R}$  to compute the distance between two point elements of spatial objects. Using this operation, we can define the notion of interval continuity for each defined interval of the interval-based spatio-temporal mapping.

**Definition 2** Let  $f : \tau(\alpha)$  be an interval-based spatio-temporal mapping and  $t$  be an element of the union of its defined intervals  $D = [t_1, t_2] \cup [t_3, t_4] \cup \dots \cup [t_{n-1}, t_n]$ .  $f$  is *upper semicontinuous* at  $t$  if  $\forall p \in f(t) \exists q \in f(t + \varepsilon) : \lim_{\varepsilon \rightarrow 0} \delta(p, q) = 0$ . Similarly,  $f$  is *lower semicontinuous* at  $t$  if  $\forall p \in f(t) \exists q \in f(t - \varepsilon) : \lim_{\varepsilon \rightarrow 0} \delta(p, q) = 0$ . We say that  $f$  is *interval continuous* if it is upper semicontinuous at every  $t$  except when  $t \in \{t_2, t_4, \dots, t_n\}$  and lower semicontinuous at every  $t$  except when  $t \in \{t_1, t_3, \dots, t_{n-1}\}$ .

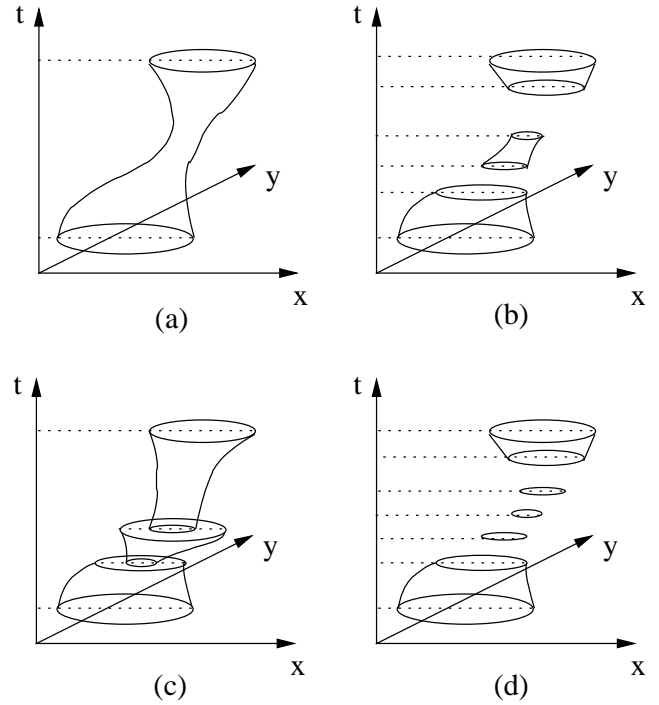
Since upper semicontinuity at  $t$  requires that  $f(t + \varepsilon)$  is defined, this concept does not apply to the right endpoints of the defined intervals. Similarly, lower semicontinuity does not apply to the left endpoints of the defined intervals. Therefore, we consider these facts in the definition of interval continuity above. If we require that the interval-based spatio-temporal mapping be interval continuous, we can now exclude instantaneous jumps (Figure 3(c)) since in such cases, the spatio-temporal mapping function is not interval continuous.

With respect to the evolution of spatial objects, we can also use the concept of interval continuity to define *continuous evolution*. Since we employ the concept of distance between point-set elements to define interval continuity, the continuous evolution of spatial objects can be thought of as continuous movements of their point-set elements inward (shrinking) or outward (growing). Therefore, interval continuity also accounts for continuous evolutions. It also excludes cases such as instantaneous shrinking or growing. Examples of valid and invalid evolutions are illustrated in Figure 4.

### 3.2 Movement and Evolution of Complex Spatial Objects

Although the concept of interval continuity of movement and evolution is reasonably clear with respect to simple, single-component spatial objects, it is not obvious how this concept can be applied to complex, multi-component spatial objects. The central issue here is to determine how a complex object moves or evolves. In this section, we provide an answer to this question using the concept of interval continuity established earlier. We define a concept of *complex continuity* on an interval-based spatio-temporal mapping to describe the movement and evolution of complex objects as follows:

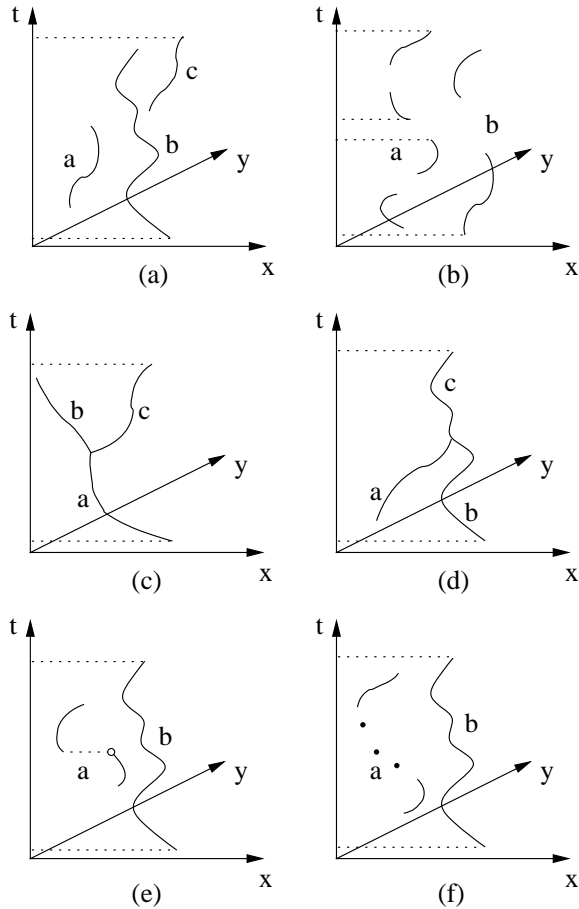
**Definition 3** A spatio-temporal mapping  $\tau(\alpha) = \text{time} \rightarrow \alpha$  where  $\alpha$  describes a complex spatial data type is called *complex continuous* if for each component of an object of  $\alpha$ , its spatio-temporal mapping is interval-based and also interval continuous.



**Figure 4: Examples of valid evolutions: a single continuous interval (a) and multiple continuous intervals (b). Examples of invalid evolutions: instantaneous shrinking within an interval (c) and instantaneous appearances (d).**

It is important to note that the continuity of a moving complex object requires the continuity of all of its components. Thus, the continuity of a complex point depends on the continuity of each of its single point components. Similarly, the continuity of a complex line relies on the continuity of each of its block components which in turn relies on the continuity of each simple line component of the block. For a complex region, its complex continuity depends on the continuity of each of its faces. As a result, complex continuity allows a complex object to move or evolve based on the movement or evolution of its components. Figure 5 illustrates valid and invalid movements for complex points.

Besides continuity, it is also interesting to investigate the interactions such as splitting and merging of components of a moving complex object. We define the *life time* of a component of a moving complex object as the smallest time interval that includes all defined intervals of the component, that is the interval  $[t_1, t_n]$ . Thus, the life time of a component may be different from the life time of other components. In fact, the life time of the entire complex object is the union of the life time of all of its components. With this concept of life time, we can now define splitting and merging of components of moving complex objects. For the case of splitting (merging), we define a *splitting instant* (*merging instant*) as a time instant which marks the end of the life time of the predecessor component (components) and the beginning of the life time of the successor components (component). With this specification, we allow splitting and merging of components of a complex object without violating the complex continuity concept of the movement. Splitting and merging are depicted in Figure 5(c)-(d).



**Figure 5: Examples of valid complex object movements and evolutions: a single continuous interval (a), multiple continuous intervals (b), splitting of a component (c), and merging of components (d). Examples of invalid movements: an instantaneous jump of a component (e) and instantaneous appearances of a component (f).**

### 3.3 Spatio-Temporal Data Types for Historical Movements of Moving Objects

By imposing the complex continuity concept as a requirement for a spatio-temporal mapping  $\tau(\alpha)$ , we can now use it to model historical movements of moving objects. By applying  $\tau(\alpha)$  to complex spatial data types *point*, *line*, and *region*, we obtain a precise definition of the corresponding moving data types *hpoint*, *hline*, and *hregion* for representing historical movements of moving objects.

$$hpoint = \tau(point) = \{f : time \rightarrow point | f \text{ is complex continuous}\}$$

$$hline = \tau(line) = \{f : time \rightarrow line | f \text{ is complex continuous}\}$$

$$hregion = \tau(region) = \{f : time \rightarrow region | f \text{ is complex continuous}\}$$

We use *hpoint*, *hline*, and *hregion* to describe historical movements of moving points, moving lines, and moving regions respectively. Due to the complex continuity constraint, these moving data types are more appropriate for representing past movements of moving objects than the existing moving data types *mpoint*, *mline*, and *mregion* shown in Section 2. Thus, we will use our new moving data

types defined above as (the *string*) part of our Balloon model described in Section 5.

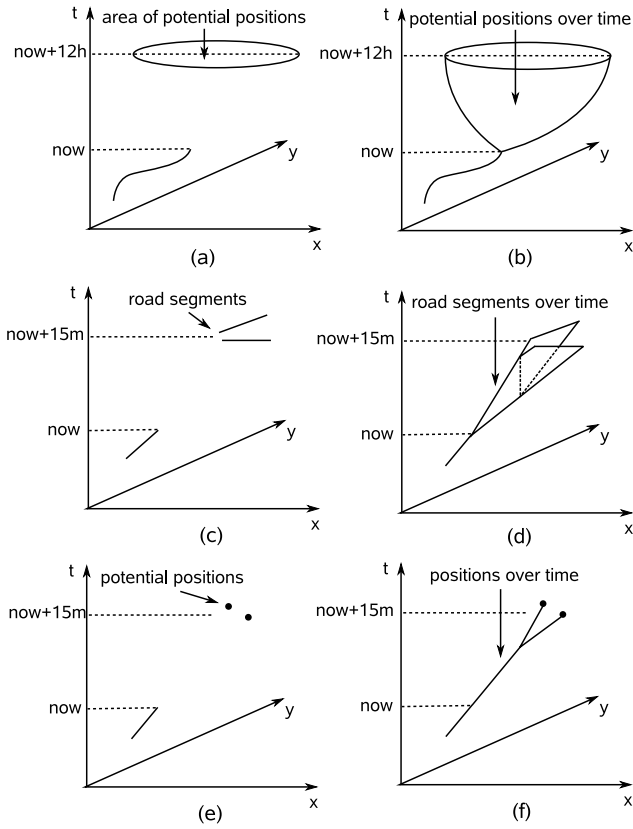
## 4. MOVING OBJECTS: MODELING FUTURE MOVEMENTS

In this section, we present our approach for modeling future movements and evolutions of moving objects. Taking into account the inherent uncertainty of the future, we explore how the future positions and extent of a moving object can be represented in Section 4.1. We present our data types for future movements of moving objects in Section 4.2.

### 4.1 Representing the Uncertainty of Future Positions and Extents of Moving Objects

Unlike movements in the past for which we (assume to) have precise knowledge, future movements involve predictions with regard to the future positions or extent of moving objects. We distinguish two different tasks in this aspect: the task of how to predict the future movements and the task of how to model the prediction data. The former deals with the development of prediction methods on how future movements are predicted. The latter deals with generic data modeling for future data. In general, prediction methods are domain and application specific. However, what we want here is a general purpose model and tool. Any determination of a prediction model would be highly restrictive and unfavorable due to its limited applicability. In fact, the development of a prediction method is typically a whole discipline by itself, and this task should belong to the domain experts. For instance, meteorologists should be responsible for developing prediction models and algorithms for weather forecasting while a database management system (DBMS) is responsible for providing storage support and querying capability for the forecasted data. We strongly believe that it is not the task of a DBMS to predict the future; however, it is imperative that a DBMS provides data modeling and persistence support for storing and querying prediction data. Based on this idea, we are only interested in the data modeling aspect of the future movements of moving objects and how this type of movements can be represented.

To take into account the uncertainty of future movements, we consider modeling the set of potential future positions or extent of a moving object. For example, the position of the eye of a hurricane at 12 hours in the future may possibly be anywhere within a predicted region (Figure 6(a)). This region represents the set of potential future positions of the hurricane's eye at 12 hours in the future. Likewise, if we are interested in the future positions for a certain period in the future, say from the present time to 12 hours in the future, then it can be anywhere within a predicted volume (Figure 6(b)). This volume can be thought of as a development of a predicted region over a period of time. Hence, it is a moving region which can be represented by using our moving data type *hregion*. Similarly, we can also use a line to represent the segments of road potentially containing the future position of a vehicle at 15 minutes in the future (Figure 6(c)). For a period between the present and 15 minutes in the future, we can use a moving line (*hline*) to represent the vehicle's potential movements (Figure 6(d)). If the vehicle always maintains a constant speed, we can also use a point object to represent its potential positions at 15 minutes in the future (Figure 6(e)). Consequently, a moving point (*hpoint*) can be used to represent its potential movements for the 15 minutes period (Figure 6(f)). Based on the same idea, if we consider an object with extent, for example a line object, its potential movement (extent) can be described by either a moving line or a moving region depending on



**Figure 6: The set of potential future positions: for a hurricane’s eye at 12 hours (a) and within the 12-hour period (b), for a vehicle at 15 minutes (c) and within the 15-minute period (d), for a vehicle with constant speed at 15 minutes (e) and within the 15-minute period (f).**

the environment and the nature of its movement. For a region object, its future movement over time can be represented by a moving region.

In summary, the set of potential future positions or the extent of a moving object can be modeled using a spatial object for an instant of time or a moving object for a period of time in the future. However, this concept does not specify the relative chance or degree of confidence with which a potential future position will eventually be the position of the moving object. To do this, we require a concept of *confidence distribution* ( $C$ ) such that each potential future position is associated with a degree of confidence. At a future time instant, we can now model the set of potential positions or the extent of an object by imposing a confidence distribution on a spatial object representing the future positions. We define  $C$  in Definition 4.

**Definition 4** The confidence distribution  $C(\alpha)$  of a spatial object of type  $\alpha$  representing the set of future positions or extent of a moving object is defined as  $C(\alpha) = \alpha \rightarrow [0, 1]$  such that the degree of confidence of every point in  $\alpha$  is between 0 and 1.

The above definition allows any type of distribution for  $C(\alpha)$ . In fact,  $C(\alpha)$  can also be a fuzzy set describing the level of membership for each point in  $\alpha$ . To apply this concept of confidence

distribution to a moving object for representing future movements over time, we can use the concept of spatio-temporal mapping to temporally lift  $C$  over time to define a *moving confidence distribution* ( $MC$ ) as shown in Definition 5.

**Definition 5** The moving confidence distribution  $MC(\alpha)$  of a spatial object of type  $\alpha$  representing the set of future positions or extent of a moving object is defined as  $MC(\alpha) = time \rightarrow C(\alpha) = time \rightarrow (\alpha \rightarrow [0, 1])$ .

Here, we do not restrict how  $MC(\alpha)$  develops over time since the confidence distribution information is domain specific, thus it may take any shape or form according to  $\alpha$  depending on a given prediction model. To adequately describe the future movement of a spatial object, we need two objects. The first object is a moving object representing the temporal evolution of a spatial object in the future. The second object is a moving confidence distribution expressing the uncertainty of the movement.

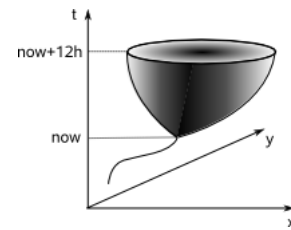
To illustrate the concepts presented, consider the example of a hurricane. We can model the set of potential positions of the eye of this hurricane using an *hregion* object (Figure 6(b)). By applying a moving confidence distribution on the *hregion* object, we obtain a new kind of object which represents the set of potential future positions, each with a degree of confidence, of the hurricane’s eye (Figure 7). This means that the future movement trajectory of the hurricane’s eye is predicted to be a part of this object. We define the data types for this kind of object in the next section.

## 4.2 Spatio-Temporal Data Types for Future Movements of Moving Objects

Provided that a combination of a moving object and a moving confidence distribution can be used to represent the set of potential future movements of a moving object, we can use this concept to define our future movement data types. Thus, a future movement data type can be constructed using the type constructor  $\varphi$  defined below.

**Definition 6** Let  $\alpha$  be a spatial data type and  $MC(\alpha)$  be a moving confidence distribution on  $\alpha$ . The future movement data type  $\varphi(\alpha)$  is defined as  $\varphi(\alpha) = \tau(\alpha) \times MC(\alpha)$  such that for each  $f = (m, p) \in \varphi(\alpha)$ ,  $m$  is defined whenever  $p$  is defined.

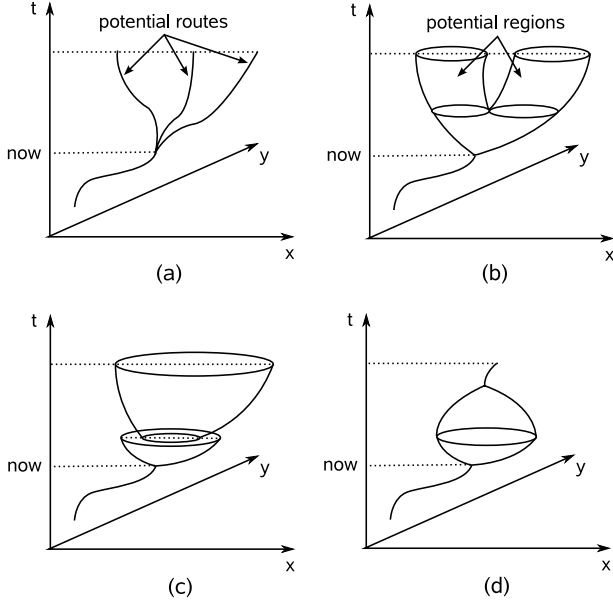
Applying this definition to the spatial data types *point*, *line*, and *region*, we obtain three future movement data types future point (*fpoint*), future line (*fline*), and future region (*fregion*):



**Figure 7: Representing the future movement of a hurricane’s eye using a moving region with a moving confidence distribution. The gradient indicates varied degree of confidence.**

$$\begin{aligned}
fpoint &= \varphi(\text{point}) = \tau(\text{point}) \times MC(\text{point}) \\
fline &= \varphi(\text{line}) = \tau(\text{line}) \times MC(\text{line}) \\
fregion &= \varphi(\text{region}) = \tau(\text{region}) \times MC(\text{region})
\end{aligned}$$

Since each future data type is defined partially using our moving data type  $\tau(\alpha)$ , its validity also depends on the validity of the moving object represented by  $\tau(\alpha)$ . Examples of valid and invalid future movements are depicted in Figure 8.



**Figure 8: Examples of valid future movements: an *fpoint* object (a) and an *fregion* object (b). Examples of invalid future movements: a discontinuous *fregion* object (c) and a dimensional collapsed *fregion* object (d).**

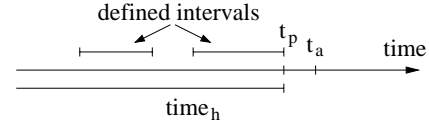
It is important to understand that these future data types are defined only for future movements of moving objects. They do not make any reference or assumption on the historical moving data types of the moving objects. For example, an object of type *fregion* can be used to represent the future movement of either a moving point, a moving line, or a moving region. To model the nature of movement of a moving object properly, we require both the past model and the future model. We will see in the next section that not all combinations of the past and future data types represent valid movements.

## 5. THE BALLOON MODEL

In this section, we present our Balloon model which is the integration of our historical and future movement models. Generally speaking, to build a balloon, we first need to prepare the string and the body of the balloon. We do this for our historical model and future model in Section 5.1 and 5.2 respectively. Once all the parts are ready, they can then be connected to form a balloon. We perform this step by integrating both of our past and future models to construct the Balloon model and define the data types in Section 5.3. Due to space limitations, we can only provide the signature of a subset of the operations that are available as part of this model in Section 5.4.

## 5.1 Preparing the String: The Temporal Domain of Historical Movements

In Section 3, we have defined our model for historical movements resulting in a type constructor  $\tau(\alpha) : \text{time} \rightarrow \alpha$  such that the spatio-temporal mapping is complex continuous. Since we use this type constructor to construct data types for historical movements, all time intervals specified as the domain of these data types must be intervals in the past. The latest known state of a moving object is assumed to be the *present state* of the object. This means that the present state of the object changes for every update of the object's position. Each update can either signify a continual movement of the last interval of knowledge or a period of no knowledge followed by a new period of knowledge. In any case, the present state of the object is always defined as the state of the object at the ending instant of the last interval of knowledge. We denote the present state of an object by  $t_p$ . It is possible that  $t_p$  may be older than the *absolute present* (the current clock time), denoted by  $t_a$ . This situation can happen when we do not have information about the state of the object at the absolute present (Figure 9). On the other hand, it is impossible for  $t_p$  to be younger than the absolute present. This would mean that the object has already moved in the future which is impossible. Hence, it is required that  $t_p \leq t_a$  holds for all moving objects. We now define a temporal domain  $\text{time}_h$  for the historical movement of a moving object as a half-infinite time domain that starts from  $-\infty$  and ends at  $t_p$  inclusively. Thus,  $\text{time}_h = (-\infty, t_p]$ . We choose to end  $\text{time}_h$  at  $t_p$  instead of  $t_a$  because if  $t_p < t_a$ , we may want to predict the movement between  $t_p$  and  $t_a$  which is outside of  $\text{time}_h$ . Otherwise, this prediction would be a part of  $\text{time}_h$  which contradicts our definition above. The temporal domain  $\text{time}_h$  is applied to all data types representing the past movement of a moving object. It will also be used in Section 5.3 as part of a temporal composition to compose the entire temporal domain of a balloon object. Figure 9 illustrates an example of  $\text{time}_h$  and  $t_p$  for a moving object.



**Figure 9: An example of a historical temporal domain.**

## 5.2 Preparing the Body: The Temporal Domain of Future Movements

For future movements of moving objects, we have defined a type constructor  $\varphi(\alpha) : \tau(\alpha) \times MC(\alpha)$  which accepts as a parameter a spatial type representing the set of future positions or the extent of a moving object. Here, we would like to restrict the temporal domain of the future data types produced by  $\varphi$  such that these data types describe only the future movements of moving objects. We define a temporal domain  $\text{time}_f$  for future movements of a moving object as a half-infinite open interval of time which starts exclusively from  $t_p$  and extends indefinitely towards  $+\infty$ . Hence,  $\text{time}_f = (t_p, +\infty)$ . We choose to begin  $\text{time}_f$  from  $t_p$  because this would allow a prediction to be made as close to the latest known state of the object as desired irrespective to the value of the absolute present. This temporal domain is applicable to all objects of both  $\tau(\beta)$  and  $MC(\beta)$ . Figure 10 depicts an example of  $\text{time}_f$  for a moving object.

## 5.3 The Balloon Model and Balloon Data Types

Having defined  $\text{time}_h$  and  $\text{time}_f$  to restrict the temporal domains of historical and future movements of a moving object, we are now

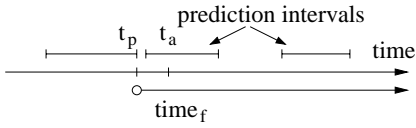


Figure 10: An example of a future temporal domain.

ready to define our Balloon model. We integrate the past ( $\tau$ ) and the future ( $\varphi$ ) type constructors to form a new type constructor  $\Omega$  for balloon objects which is defined on the entire time domain. This time domain is a result of a temporal composition of  $time_h$  and  $time_f$ . Since the spatial type as the argument of  $\tau$  refers to a different spatial object than that of the spatial type as the argument of  $\varphi$ , we denote the former by  $\alpha$  and the latter by  $\beta$ . Thus, the type constructor  $\Omega(\alpha, \beta)$  is equal to  $\tau(\alpha) \times \varphi(\beta)$ . That is, for a balloon object  $b = (h, f)$ ,  $h \in \tau(\alpha)$  represents the past movement of a spatial object of type  $\alpha$  and  $f \in \varphi(\beta)$  represents the future movement of  $b$  given that the set of its potential future positions at a future time instant is described by a spatial object of type  $\beta$ . As indicated earlier, not all combinations of  $\alpha$  and  $\beta$  constitute valid movements. For example, it is not possible to use  $fpoint$  to represent the future extent of a moving region. If this were to be possible, this means that, at a future time instant, the region object would have evolved into a point object. This is not possible since our definition of complex continuity does not allow movements involving dimensional collapse. Therefore, the combination for which  $\alpha$  is the region type and  $\beta$  is the point type has been proved to be an invalid combination. The general idea here is that the set of potential future positions or extent of a moving object at a future time instant must be a spatial object of dimension greater than or equal to the dimension of the object being moved. This means that an object can move or evolve such that at a future time instant, it fits in or equal to its prediction without collapsing its dimension. Let  $dim$  be a function that returns the dimension of a spatial type. It is required that  $dim(\beta) \geq dim(\alpha)$  holds for all valid combinations of  $\alpha$  and  $\beta$ . We now define our type constructor  $\Omega$  for balloon data types in Definition 7.

**Definition 7** The type constructor  $\Omega(\alpha, \beta)$  for a balloon data type describing a balloon object whose past movement and future prediction are based on the spatial type  $\alpha$  and  $\beta$  respectively, is defined as  $\Omega(\alpha, \beta) = \tau(\alpha) \times \varphi(\beta)$  such that all of the following conditions hold: (i)  $dim(\beta) \geq dim(\alpha)$ ; (ii)  $\tau(\alpha)$  represents the past movement and is defined on  $time_h$ ; (iii)  $\varphi(\beta)$  represents the future movement and is defined on  $time_f$ .

In other words,  $\Omega(\alpha, \beta)$  is a total function defined on the complete time domain such that it evaluates  $\tau(\alpha)$  for an instant  $t$  in  $time_h$  or  $\varphi(\beta)$  for  $t$  in  $time_f$ . By specifying  $\alpha$  and  $\beta$ , we obtain six data types for balloon objects:

$$\begin{aligned}
balloon\_pp &= \Omega(point, point) = hpoint \times fpoint \\
balloon\_pl &= \Omega(point, line) = hpoint \times fline \\
balloon\_pr &= \Omega(point, region) = hpoint \times fregion \\
balloon\_ll &= \Omega(line, line) = hline \times fline \\
balloon\_lr &= \Omega(line, region) = hline \times fregion \\
balloon\_rr &= \Omega(region, region) = hregion \times fregion
\end{aligned}$$

Each of these balloon data types consists of a data type for past movements (the strings) and a data type for future movements (the bodies). Here, we use our data types defined in earlier sections for

this purpose. Now we can use these balloon data types to model balloon objects. A balloon object has the advantage of being able to capture both the past movement and the future movement of a moving object while maintaining the consistency of the object at all time. Examples of a *balloon\_pp*, a *balloon\_pl*, and a *balloon\_pr* object are shown in Figure 8(a), 6(d), and 8(b) respectively.

## 5.4 Some Operations on Balloon Objects

Since all balloon objects are defined on the complete time domain as opposed to only the past or the future, many new operations are now available in addition to all existing operations on moving objects. In the followings, we list some of these new operations introduced by the Balloon model. Let  $\lambda$  be a spatial data type.

$$\begin{aligned}
construct &: \tau(\alpha) \times \tau(\beta) \times MC(\beta) \rightarrow \Omega(\alpha, \beta) \\
update &: \tau(\alpha) \times \tau(\beta) \times MC(\beta) \rightarrow \Omega(\alpha, \beta) \\
past &: \Omega(\alpha, \beta) \rightarrow \tau(\alpha) \\
future &: \Omega(\alpha, \beta) \rightarrow \tau(\beta) \times MC(\beta) \\
temporal\_selection &: \Omega(\alpha, \beta) \times interval \rightarrow \Omega(\alpha, \beta) \\
present &: \Omega(\alpha, \beta) \rightarrow time \\
balloon\_life &: \Omega(\alpha, \beta) \rightarrow interval \\
past\_period &: \Omega(\alpha, \beta) \rightarrow real \\
future\_period &: \Omega(\alpha, \beta) \rightarrow real \\
known\_period &: \Omega(\alpha, \beta) \rightarrow real \\
unknown\_period &: \Omega(\alpha, \beta) \rightarrow real \\
state\_at &: \Omega(\alpha, \beta) \times time \rightarrow \alpha \\
prediction\_at &: \Omega(\alpha, \beta) \times time \rightarrow \beta \\
confidence\_at &: \Omega(\alpha, \beta) \times point \times time \rightarrow real \\
past\_projection &: \Omega(\alpha, \beta) \rightarrow \lambda \\
future\_projection &: \Omega(\alpha, \beta) \rightarrow \lambda
\end{aligned}$$

The operation *construct* builds a balloon object given three parts: the past movement, the future movement, and the corresponding moving confidence distribution. A balloon object can be updated by expanding it by a new interval of movement and optionally the corresponding new prediction. If the new interval of movement starts at the present state of the object, this interval signifies a continual movement from the last interval of the object. Otherwise, an interval of unknown movement occurs preceding the new interval of knowledge. When a balloon object is updated, the present state is updated and checked against the existing prediction. If the present state is younger than the start of the prediction, this prediction is no longer applied. Thus the update operation always maintains the temporal consistency of the balloon object. The *past* and *future* operations retrieve the past and future movements respectively. To retrieve an object whose life time starts at a specified instant in the past and ends at a specified instant in the future, the *temporal\_selection* operation can be used. The present time of an object is given by the *present* operation. The life time of a balloon object is defined starting from the beginning of its past movement to the end of its prediction and is given by the operation *balloon\_life*. The *past\_period* and *future\_period* operations return the duration of the past movement and future prediction respectively. Similarly, the *known\_period* and *unknown\_period* operations determine the sum of all intervals of knowledge and the sum of all intervals of lack of knowledge within the balloon life time respectively.

In regard to the position or shape of an object, the state of a balloon object at any time instant in the past is given by the *state\_at* operation. For the future, the *prediction\_at* operation retrieves a



spatial object representing the set of potential positions or extent of a balloon object at the specified time instant in the future. With an additional point parameter, the *confidence\_at* operation can determine the degree of confidence for this point at a time instant in the future. This operation first determines the confidence distribution at the specified time instant and then applies the coordinates of the point to the distribution to obtain the result. The *past\_projection* and *future\_projection* operations provide a spatial projection of the past movement and the future prediction respectively. A spatial projection of a moving object describes only the movement in space and ignore the movement in time. Hence, the result of a spatial projection operation is a spatial object.

One of the main benefits for using the Balloon model is that we can have operations that operate on the entire time domain of a moving object. Since the Balloon model maintains the consistency of the object, each operation can be easily implemented using balloon objects as arguments without having to concern about consistency issues.

## 6. QUERYING USING THE BALLOON MODEL

For the purpose of using balloon objects in queries, the presented balloon data types can be embedded into any extensible DBMS data model as attribute data types. The operations can be registered in the DBMS such that they are available for use in queries. For example, we can integrate these data types into the relational model and construct relations for hurricanes and cities. We model the movement of a hurricane's eye by a *balloon\_pr* object. For the extent of the hurricane force wind, we model its movement using a *balloon\_rr* object. For cities, we represent their position by a point object. Hence, we have the following relations:

```
hurricanes(name:string, eye:balloon_pr,
           extent:balloon_rr)
cities(name:string, position:point)
```

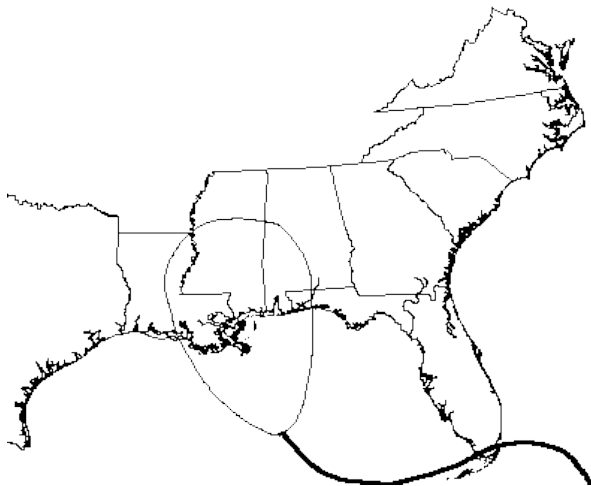


Figure 11: The movement of the eye of hurricane Katrina.

Assuming that these relations have been previously populated with all necessary data. For the purpose of our example, assume also that hurricane Katrina is currently making its way across the Gulf of Mexico (Figure 11). Before we get into some complex queries,

it may be interesting to determine how much information we currently have on Katrina. We can post a query to retrieve the life span of Katrina's eye "Give me the life span including the current prediction of the eye of hurricane Katrina":

```
SELECT balloon_life(eye)
FROM hurricanes
WHERE name="Katrina"
```

We can also determine the ratio between the prediction period and the life time of the eye's past movement. This information may be important to evaluate the maturity of the prediction. We use the operations *future\_period* and *past\_period* for this purpose.

```
SELECT future_period(eye) / past_period(eye)
FROM hurricanes
WHERE name="Katrina"
```

In a similar way, the ratio between all periods of no knowledge and all periods of knowledge can be determined by using the *unknown\_period* and *known\_period* operations. This ratio can help indicate the level of knowledge we have about the movement. Other interesting queries involve the state of the hurricane or its predictions. For instance, we can ask "What area will potentially be affected by the eye of hurricane Katrina at 12 hours from now?"

```
SELECT prediction_at(eye,now()+12h)
FROM hurricanes
WHERE name="Katrina"
```

Following the same trend, the total area that may be affected by the hurricane force wind at any time in the future can be determined by using the *future\_projection* operation on the *extent* attribute of the hurricane. Beyond these queries, we can also ask questions relating to degree of confidence such as "What is the chance that Katrina's eye will be on the city of New Orleans 24 hours from now?"

```
SELECT confidence_at(eye,position,now()+24h)
FROM hurricanes, cities
WHERE hurricanes.name="Katrina" AND
      cities.name="New Orleans"
```

With the use of spatial predicates *on* : *point* × *line* and *in* : *point* × *region* which determine whether a point is on a line and whether a point is in a region respectively, we can list all hurricanes that either hit New Orleans in the past or may hit it in the future.

```
SELECT hurricanes.name
FROM hurricanes, cities
WHERE cities.name="New Orleans" AND
      ( on(position, past_projection(eye)) OR
        in(position, future_projection(eye)) )
```

These queries illustrate the use of our operations with the movement of a balloon object. Other than these operations, we can also use spatio-temporal predicates between balloon objects, which are to be defined in the future. A number of approaches have been

proposed for specifying and using spatio-temporal predicates in queries. One solution is to use the *spatio-temporal query language* (STQL) [3] to support textual specifications of spatio-temporal predicates. STQL allows us to textually formulate spatio-temporal queries that involve the use of spatio-temporal predicates. To illustrate how spatio-temporal predicates between balloon objects can be used, consider the scenario of airplanes and ships which can be modeled as *balloon\_pp* objects due to their well defined routes. We can create the corresponding relations for these balloon objects as follows:

```
airplanes(flightNo:string, flight:balloon_pp)
ships(shipID:string, route:balloon_pp)
```

We may want to divert all airplanes whose flight will *potentially cross* the projected extent of hurricane Katrina. This query requires the use of a spatio-temporal predicate *potentially\_cross* between balloon objects. For the purpose of this example, we assume that this predicate exists between balloon objects of type *balloon\_pp* and *balloon\_rr*. Hence, we can pose the following query:

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

Unlike airplanes, ships may have a higher tolerance against storms. Thus, we may only want to identify all ships that have a high probability of being affected by the eye of the hurricane, so that appropriate actions can be taken to help or warn these ships. This requires a spatio-temporal predicate with a specific probability of occurrence, e.g., *most\_likely\_cross*, which can be predefined with a specific level on confidence.

```
SELECT shipID
FROM ships, hurricanes
WHERE hurricanes.name="Katrina" AND
      most_likely_cross(route, eye)
```

Another approach for using spatio-temporal predicates between balloon objects in queries is to employ the *visual query language* [5, 4]. This visual language allows a convenient and intuitive graphical specification of spatio-temporal predicates as well as supports the formulation of spatio-temporal queries with these predicates.

## 7. CONCLUSIONS

To properly model the past and future movements of moving objects, it is required that we have a clear understanding of how objects move or evolve. Further, modeling the future movements of moving objects requires that we take into account the inherent uncertainty aspect of the future. Modeling both the past and the future movements of moving objects requires that we additionally maintain the consistency of the movements. The Balloon model presented in this paper ensures this. Our contributions include the design of precise and flexible historical and future movement models for moving objects as well as the Balloon model which supports both types of movement while ensuring their consistency. As a result, we present new sets of spatio-temporal data types for all types of movements. With these data types, a new set of operations becomes available. In the future, we plan to investigate spatio-temporal operations and predicates between balloon objects. We

also plan to implement the Balloon model as part of our Spatio-Temporal Algebra (STAL) software package.

## 8. REFERENCES

- [1] M. Erwig, R.H. Güting, M. Schneider, and M. Vazirgiannis. Abstract and Discrete Modeling of Spatio-Temporal Data Types. In *ACM Symp. on Geographic Information Systems (ACM GIS)*, pages 131–136, 1998.
- [2] M. Erwig, R.H. Güting, M. Schneider, and M. Vazirgiannis. Spatio-Temporal Data Types: An Approach to Modeling and Querying Moving Objects in Databases. *Geoinformatica Journal*, 3(3):265–291, 1999.
- [3] M. Erwig and M. Schneider. Developments in Spatio-Temporal Query Languages. In *IEEE Int. Workshop on Spatio-Temporal Data Models and Languages*, pages 441–449, 1999.
- [4] M. Erwig and M. Schneider. Visual Specifications of Spatio-Temporal Developments. In *15th IEEE Symp. on Visual Languages (VL)*, pages 187–188, 1999.
- [5] Martin Erwig, Martin Erwig, and Markus Schneider. Query-By-Trace: Visual Predicate Specification in Spatio-Temporal Databases. In *The Fifth Working Conference on Visual Database Systems*, volume 168, pages 199–218, 2000.
- [6] L. Forlizzi, R.H. Güting, E. Nardelli, and M. Schneider. A Data Model and Data Structures for Moving Objects Databases. In *ACM SIGMOD Int. Conf. on Management of Data*, pages 319–330, 2000.
- [7] K. Hinrichs G. Trajcevski, O. Wolfson and S. Chamberlain. Managing Uncertainty in Moving Objects Databases. *ACM Trans. on Database Systems (TODS)*, 29:463–507, 2004.
- [8] R.H. Güting, M.H. Böhlen, M. Erwig, C.S. Jensen, N.A. Lorentzos, M. Schneider, and M. Vazirgiannis. A Foundation for Representing and Querying Moving Objects. *ACM Trans. on Database Systems (TODS)*, 25(1):881–901, 2000.
- [9] R.H. Güting and M. Schneider. *Moving Objects Databases*. Morgan Kaufmann Publishers, 2006.
- [10] Y. Guo S. Grumbach J. Chen, X. Meng and H. Sun. Modeling and Predicting Future Trajectories of Moving Objects in a Constrained Network. In *Int. Conf. on Mobile Data Management (MDM)*, page 156, 2006.
- [11] H. Mokhtar and J. Su. Universal Trajectory Queries for Moving Object Databases. In *Int. Conf. on Mobile Data Management (MDM)*. IEEE Computer Society, 2004.
- [12] N. Pelekis, B. Theodoulidis, I. Kopanakis, and Y. Theodoridis. Literature Review of Spatio-Temporal Database Models. *Knowledge Engineering Review*, 2005.
- [13] R. Praing and M. Schneider. A Universal Abstract Model for Future Movements of Moving Objects. In *AGILE Int. Conf. on Geographical Information Systems*, 2007.
- [14] M. Schneider and T. Behr. Topological Relationships between Complex Spatial Objects. *ACM Trans. on Database Systems (TODS)*, 31:39–81, 2006.
- [15] A.P. Sistla, O. Wolfson, S. Chamberlain, and S. Dao. Modeling and Querying Moving Objects. In *Int. Conf. on Data Engineering (ICDE)*, pages 422–432, 1997.
- [16] E. Tossebro and R.H. Güting. Creating Representations for Continuously Moving Regions from Observations. In *Int. Symp. on Advances in Spatial and Temporal Databases*, pages 321–344, 2001.