# Query operations for moving objects database systems

José Moreira
Computer Science
Department
Universidade Portucalense
Portugal
jmoreira@upt.pt

Cristina Ribeiro
Department of Electrical
Engineering and Computing
FEUP
Portugal
mcr@fe.up.pt

Talel Abdessalem
Computer Science
Department
ENST de Paris
France
talel.abdessalem@enst.fr

## ABSTRACT

Geographical Information Systems were originally intended to deal with snapshots representing a single state of some reality but there are more and more applications requiring the representation and querying of time-varying information. This work addresses the representation of moving objects on GIS.

The continuous nature of movement raises problems for representation in information systems due to the limited capacity of storage systems and the inherently discrete nature of measurement instruments. The stored information has therefore to be partial and does not allow an exact inference of the real-world object's behavior. To cope with this, query operations must take uncertainty into consideration in their semantics in order to give accurate answers to the users.

The paper proposes a set of operations to be included in a GIS or a spatial database to make it able to answer queries on the spatio-temporal behavior of moving objects. The operations have been selected according to the requirements of real applications and their semantics with respect to uncertainty is specified. A collection of examples from a case study is included to illustrate the expressiveness of the proposed operations.

## Categories and Subject Descriptors

H.2.3 [**Database management**]: Languages—*Query Languages*; H.2.8 [**Database management**]: Database Applications—*Spatial databases and GIS*

## Keywords

Spatio-temporal databases, spatio-temporal uncertainty, moving objects, movement operations

## 1. INTRODUCTION

Moving objects database systems are systems designed to store and manipulate information about the behavior of moving objects, together with spatial information representing static geographical objects, temporal information and traditional numeric and alphanumeric information. This work focuses moving objects that can be represented by points. There are many applications, such as those dealing with the position of cars, planes, maritime vessels or animals, for which the size and shape of objects is not relevant in the scale of the maps used and only their position is required.

A moving object may change location at every instant. Two problems arise when dealing with such continuously changing information: on the one hand, computer systems are not able to store and manipulate infinite sets; on the other, measurement instruments are inherently discrete and are not able to continuously capture the location of moving objects. Thus, the knowledge about the movement of an object, as it is stored in a computer system, has intrinsic limitations. It is only a partial representation and the complete knowledge of the real-world behavior of moving objects cannot be exactly inferred from the stored representation.

In this paper, we use the fact that there are physical constraints on the movement of real world objects to effectively deal with partial knowledge. We use an approach to the representation of moving objects that allows ascertaining their position within a known precision. Based on this, we propose a set of operations covering the sort of queries typically performed on the spatio-temporal behavior of moving objects. Each operation is associated to an intended semantics that copes with uncertainty.

The paper is organized as follows. Section 2 describes a method for bounding uncertainty on the location of moving objects and presents the different semantics that can be attached to a base operation in order to deal with uncertainty. Section 3 presents the movement operations in several categories. The expressiveness of the proposed framework is analyzed in section 4 using examples from a real-world case study. Sections 5 and 6 present an overview of related work and directions of current and future research.

## 2. MOVEMENT ISSUES

Sensor systems, like GPS, are only able to get information at discrete points, and thus the exact behavior of moving objects between any two consecutive observations is unknown. Consider the case of a port authority dealing with a spread of toxic waste in the sea and querying a nautical surveillance system to know which ships have crossed the polluted zone for a specified time interval. Let us consider that the ship responsible for the waste has actually followed the trajectory represented in figure 1. The black dots represent four observations made during the specified time period, the shaded rectangle represents the polluted area and the hatched line a trajectory that might have been inferred from the observations.

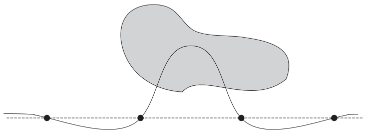

**Figure 1: Indeterminacy of the behavior of an object between consecutive observations**

The hatched line does not cross the shaded region and, thus, an answer to the query based on this estimation of the trajectory would not include the guilty ship. On the contrary an answer may include false candidates whose inferred trajectory crosses the area even though they have not actually been there.

## 2.1 Bounding uncertainty

There are physical constraints on the movement of objects, namely the maximum velocity of an object, allowing limiting uncertainty of their position [9].

Let's consider a moving point $m$, a time instant $t$ in an interval $\langle t_1, t_2\rangle$ between two consecutive observations $t_1$ and $t_2$, and two variables $\Delta t_1 = t - t_1$ and $\Delta t_2 = t_2 - t$. $p_1$and $p_2$ correspond to the position of the moving object at the observation time instants $t_1$ and $t_2$, respectively. At time $t$, the distance $d_1$ between $m$ and $p_1$ is inferior to $r_1 = vv_{\max} \times \Delta t_1$ and the distance $d_2$ from $p_2$ is inferior to $r_2 = vv_{\max} \times \Delta t_2$, where $vv_{\max}$ is an user-defined value standing for the maximum velocity for a moving object (figure 2).

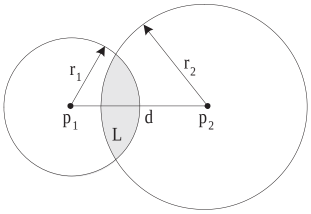

**Figure 2: A *lens area* for a time instant**

At time $t$, the moving object might be at any location within the area defined by the intersection of the two circles. This is a so-called *lens* area [12] representing the set of all possible locations for a moving object at a certain time. Uncertainty is not uniform: it ranges from no uncertainty, at time instants corresponding to observations, to a maximum uncertainty corresponding to the middle point between observations.

The set of all locations where a moving object might be between two consecutive observations corresponds to an ellipsis (Figure 3) and the parameters $a$ and $b$ may be computed as follows: $a = vv_{\max} \times (t_2 - t_1)/2$, $c = p_2 - p_1/2$ and $b^2 = a^2 - c^2$.

Figure 4 shows how to compute the lens area for a time interval $\langle t_a, t_b\rangle$ between two consecutive observations.

The circle with radius $r_b = vv_{\max} \times (t_b - t_1)$ corresponds to the maximum distance from $p_1$ that could be reached by the moving object at $t_b$. The same reasoning applies to $r_a$. In addition, the moving object could not have been outside the ellipsis just defined. The lens area is the one defined by the intersection of the two circles with the ellipsis. Lens areas for time intervals comprising one or more observations are the union of several lens areas computed using the method described above.

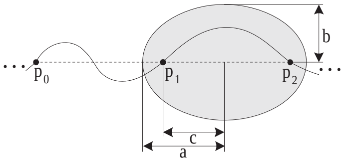

**Figure 3: Coverage of all possible trajectories between two consecutive observations**

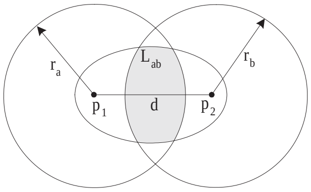

**Figure 4: A *lens area* for a time interval between consecutive observations**

## 2.2 Semantics

To integrate moving objects into a GIS or a spatial database system it is necessary to establish the relevant operations to be used in queries and appropriate semantics to interpret them and to deal with uncertainty.

Data collected by sensor systems allows estimating the location of observed objects at any time instant between observations, assuming, for example, that the movement is linear and uniform between two consecutive observations. This semantics is not satisfactory to answer queries where uncertainty is significant. Hence, we propose to combine basic operations on moving objects with different semantics to provide meaningful operations in this context.

### 2.2.1 Possibly Semantics

We add the prefix *Possibly* to operations concerned with returning the set of all possible candidates matching some predicate. Answers to such queries are supersets of the ideal results obtainable in an infinite precision representation. If the question is "Which are the objects that have been in Area C", the answer includes all objects that actually have been in that area and it may also include some objects that have not. Similarly, if the question is "Show me the path of object O within Area C", the answer includes all parts of the movement of object O for which it actually was within Area C and it may also include some parts for which the object was not there.

The set complement of this result consists in the values that definitely do not match the query predicate.

### 2.2.2 Surely Semantics

We add the prefix *Surely* to operations returning only values that definitely match a predicate. Answers using such operations are subsets of ideal results. If the question is "Which are the objects that have been in Area C", the answer includes only objects that actually have been in Area C and it may omit some objects that have been in the area as well. If the question is "Show me the path of object O within Area C", the answer includes only parts of the movement of object O for which it was actually within Area C and it may exclude some parts for which object O also was within Area C.

The set difference of the result of an operation using the Possibly semantics with the result of the same basic operation using the Surely semantics returns a set of values for which it is neither possible to assert that they do match the query predicate nor that they do not.

### 2.2.3 *Probably Semantics*

Operations with prefix *Probably* consider the existence of methods that allow estimating the location for a moving object at any time instant. The result of the estimation is a probable location. Answers to queries using these operations are sets of possible candidates. They may include some false hits and they may also be incomplete.

# 3. MOVEMENT OPERATIONS

The semantic variants proposed above are only meaningful in movement related operations. This is the main reason for integrating these semantics at the level of the operations rather than introducing them more generically at the query language level.

Throughout this exposition, we assume the existence of an infinite set of constants $D$ for the various data types required, with the following identified subsets: $N = \{n_1, n_2, \ldots\}$ as numeric constants for integers and real numbers; $T = \{t_1, t_2, \ldots\}$ as temporal constants for time instants and time intervals; and $G = \{g_1, g_2, \ldots\}$ as geographical constants for points, lines and polygons.

We also define the *movement* of an object as an infinite set of elements from $T \times G$. A *movement value* is a finite representation of a movement. $Mvt = \{m_1, m_2, \ldots\}$ denotes the set of movement values. For definitions of finite representations of movement see our previous work [9] or [15].

The sets $N$, $T$ and $G$ suffice for introducing any movement operation. Notice that temporal and spatial values are explicitly represented in a movement. Numeric values, like speed and orientation, are implicit, and may be inferred from the former. Any other data types, like strings, are irrelevant for movement operations. Simply lifting non-temporal operations so that they work on movement values is not sufficient to capture the whole diversity of real-world phenomena. As we will see a novel category of operations, which we refer to as spatio-temporal operations, emerges in this context. The other ones are the typical projection, restriction and metric operations.

## 3.1 Projection operations

Projections are operations of the kind $Mvt \longrightarrow D$, used to extract temporal or spatial parts or to evaluate numeric quantities associated with a movement.

### 3.1.1 *Temporal projection*

A temporal projection returns the temporal coverage of a movement composed by time instants or time intervals. Gaps are allowed in movements, to cope with periods of time where the position of objects is unavailable or not required. There is just one operation in this class, named *TemporalProjection*.

### 3.1.2 *Spatial projection*

A spatial projection returns the spatial coverage of a movement, i.e., the path followed by an object. There are two operations classified as spatial projections: *SpatialPathProjection*, returning a line representing an estimation of the path followed by a moving object, and *SpatialLensProjection*, returning a lens area corresponding to the set of all locations that might be attained by a moving object during its movement.

### 3.1.3 *Numeric projection*

A numeric projection extracts numeric values associated with a movement. The operations considered in this class are intended to deal with the object's velocity vector. Two operations are considered as numeric projections. *Speed* returns an approximation of the average speed of a movement, within the constraints of available information. *Orientation* returns the angle of the vector traced from first to last positions in a movement. In both cases, the accuracy of the values obtained can only be estimated at application level.

## 3.2 Restriction operations

Restriction operations are intended to filter movement values according to some criteria. The result is one or several parts of the initial movement, for which the criteria hold.

### 3.2.1 *Temporal restriction*

A temporal restriction is a function in the form $Mvt \times T \longrightarrow Mvt$ and filters a movement $m \in Mvt$ according to a temporal element $t \in T$. We propose the following operations in this class: *Before*, *During*, *After* and *At*. The arguments for the *During* operation are a movement and a temporal element. The temporal argument of the other operations is a time instant. Notice that the *During* operation is sufficient to filter any part of a movement according to some temporal restriction. The others are included just for simplicity of language.

### 3.2.2 *Spatial restriction*

According to [2], spatial relationships in GIS or spatial databases are grouped in three categories: topological, directional and distance. The same classification holds for the relationships between moving and static spatial objects and between moving objects. The distinction is that spatial relationships involving moving objects are temporary, i.e. they may hold only for certain time periods. Moreover, we also want to consider uncertainty in the location of moving objects. The general format is $Mvt \times X \longrightarrow Mvt$ for topological and directional operations and $Mvt \times X \times N \longrightarrow Mvt$ for distance operations, where $X$ stands for either $G$ or $Mvt$.

**Topological relationships** do not change with transformations of scale, translation and rotation. According to the Calculus-based method [1] there are three topological relationships to be considered between a point and a static geographical object: in, touch and disjoint. A moving point $m$ is considered to be in a static object $g$ when it intersects its interior, to touch the object when it intersects its boundary and to be disjoint from the object when it intersects its exterior. The boundary of a line consists of its end points and the interior consists of the linear non-intersecting connection between them; for a point there is no distinction between boundary and interior. The topological relationships that may hold between two moving objects represented by points are disjoint and intersect [1]. Two moving objects intersect when they are at the same location at the same time and are disjoint otherwise.

Denoting by $l$ the lens area of a moving object $m$, $m$ is possibly in a geographical object $g$ when $l$ intersects the interior of $g$, $m$ is possibly touching $g$ when $l$ intersects the boundary of $g$ and $m$ is possibly disjoint from $g$ when $l$ intersects the exterior of $g$. Similarly, $m$ is surely in $g$ when $l$ is contained in $g$; $m$ is surely disjoint from $g$ when $l$ intersects neither the boundary nor the interior of $g$. To assert that $m$ is surely touching $g$ there must exist an observation taken when $m$ was on the boundary of $g$.

A moving object possibly intersects another moving object when their lens areas have some part in common. They are surely disjoint otherwise. Two moving objects surely intersect when they have been observed at the same time and place. They are possibly disjoint otherwise.

It is apparent that the results of disjoint and intersect operations are complementary and thus we only maintain the disjoint relationship, which has already been defined for relationships between moving objects and static spatial objects.

The nine spatial restriction operations between a moving and a static geographical object are therefore of the form: *Pref-In*, *Pref-Touch* and *Pref-Disjoint*, such that *Pref-* stands for *Probably*, *Possibly* or *Surely*. For spatial restrictions between two moving objects, only *Pref-Disjoint* operations are applicable.

**Direction relationships** deal with relative order in space. There is currently no well-established definitions for them. Many authors follow a projection-based approach where direction relationships are defined using projection lines orthogonal to the coordinate axes [2, 11]. The main relationships are north, south, east and west, and composite relationships such as northeast are constructed by conjunction of previous ones.

A moving object is at north of a static geographical object $g$ when its y-coordinate is above the upper bound of the y-coordinate of $g$. Taking uncertainty into account, the y-coordinate is replaced by the upper (possibly semantics) or the lower (surely semantics) bounds of the lens area associated with the moving object. A similar approach is applied to directional relationships between two moving objects.

Combining the four basic directions with the three uncertainty semantics there are twelve directional operations: *Pref-North*, *Pref-South*, *Pref-East* and *Pref-West*, where *Pref-* stands for either *Probably*, *Possibly* or *Surely*.

**Distance relationships** express properties reflecting the concept of a metric. They are invariant under translation and rotation but they change under scaling.

Restrictions based on distance values allow selecting parts of the movement of an object for which the distance to a another object is at most equal to a numeric value.

The distance between a moving and a static object is measured between the point representing the location estimated for the former at a certain time and the closest point of the latter. For possibly and surely semantics, the closest or the farthest points of the lens area are considered, respectively, instead of the estimated location. For the distance between two moving points, the references are either the estimated locations or the closest (farthest) points of their lens areas.

We denote these operation by *Pref-WithinDistance*, where *Pref-* stands for *Probably*, *Possibly* or *Surely*.

### 3.2.3 Numeric restriction

These operations are based on the numeric values that may be inferred from a movement representation and their general format is $Mvt \times N \times N \longrightarrow Mvt$. As in the case of projections, we propose two operations: *SpeedBetween* and *OrientationBetween* that return the parts of a movement for which the average speed or angle are bounded by two given values.

## 3.3 Metric operations

To answer questions about maximum or minimum distances a two-step approach can be used. For the distance between a moving and a static object, the first step consists in obtaining the line or the lens area corresponding to the path followed by the moving object, using a spatial projection function. The second step consists in evaluating the minimum or the maximum distance between the line or the lens area and the static geographical object considered. For distances between two moving objects, the first step consists in two projection operations to obtain two lines or two lens areas and the second step consists in the evaluation of the distance between them. Notice that, in both cases, the second step consists in a spatial operation already considered in GIS.

Users may also want to obtain moving distances [8]. A moving distance is a time-varying numeric value representing distances between two moving objects or between a moving and a static object for a specified period of time. For an observation-based system as we are considering, the distance between two moving objects, or between a moving and a static object, is not only a time-varying numeric value but also a value subject to uncertainty. The complexity involved in defining this kind of operation indicates that this subject requires specific attention and is left for further work.

## 3.4 Topological-temporal operations

There are topological relationships between movements and between a movement and a static spatial object that hold for a certain time depending on the behaviour of the object on preceding and subsequent times. These relationships are enter, leave and cross and we refer to them as topological-temporal. Their general format is $Mvt \times X \longrightarrow Mvt$ where $X$ stands for either $G$ or $Mvt$.

A moving object $m$ enters a region (static object) $g$ if $m$ intersects the exterior and the interior of $g$ in this exact temporal order; $m$ leaves $g$ when intersections occur in reverse order; a cross event occurs when $m$ enters and then leaves $g$. The results for the enter and leave operations consist in the parts of $m$ for which the moving object was over the boundary of $g$; for the cross operation the result consists in the parts of $m$ between the entering and leaving events.

Taking uncertainty into account, topological-temporal relationships may hold whenever a lens area partially overlaps $g$ for a certain time. They surely hold when there have been observations inside and outside the region or when there are sequences for which the lens area was totally inside (outside) and then totally outside (inside) a region.

Although there is no distinction between interior and boundary for a point, operations enter and leave still make sense for real life applications. As an example, consider an aircraft entering an aircraft carrier, in a system where both are represented by points. The cross operation for moving objects is not considered since, so far, it has had no useful application in real life examples.

A moving object $m_1$ enters another moving object $m_2$ if they are disjoint and then they intersect, and $m_1$ leaves $m_2$ when they first intersect and then are disjoint.

Taking uncertainty into account, a moving object $m_1$ may enter or leave another moving object $m_2$ whenever the lens areas are not disjoint.

There are nine operations for the topological-temporal operations between a moving and a static object: *Pref-Enter*, *Pref-Leave* and *Pref-Cross*, where *Pref-* stands for *Probably*, *Possibly* or *Surely*. There are six topological-temporal operations between two moving objects since the cross relationship is not applicable.

## 3.5 Integration within a database system

A moving objects database should be able to store and manipulate numeric, alphanumeric, temporal, geographical and movement values. The new issue is the integration of movement into database systems, given that others are already considered in traditional, temporal or spatial database systems. Having classified movement operations into categories in previous section, some considerations should be done at this point.

The numeric projections and numeric restrictions are approximate and the accuracy of the values obtained can only be estimated at application level. Metric operations for computing moving distances, speeds or orientations, in the context of partial knowledge described in this paper, are left open for future research.

The quality of the answers that can be computed with these operations depends on a number of application level factors, such as the elapsed time between observations, the spatial precision or the velocity of the represented objects. Ideally, the difference between the results of operations under possibly semantics and those obtained under surely semantics should be minimized.

Figure 5 illustrates how movement operations can be integrated with the other operations in a database system.

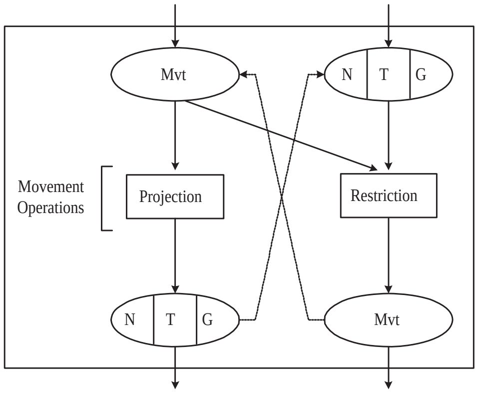

**Figure 5: Integration of movement operations in a database**

Movement values are the entries for movement projections and restrictions. These values may be obtained from data stored in the database or from the results of other movement restriction operations. Numeric, temporal and geographical values are entries for movement restrictions and are used as criteria to filter movement values. They may be obtained from data stored in the database, the result of movement projection operations, the result of other operations over non-movement values or from user entries.

The result of movement operations is stored in the database or displayed to the user. Moreover, the results of movement restrictions may be used as entries for another movement operation (projection or restriction) whereas the results of movement projections may be used as entries for movement restrictions or for operations dealing with non-movement values.

## 4. EXAMPLES OF MOVEMENT OPERATIONS

This section presents some examples to illustrate the expressiveness of the operations proposed. We use as case study, the MONICAP system for monitoring and control of fishing activities (CCMP, Inesc). The system is being used since 1992 by the Portuguese general authority for fishing activities (IGP). It continuously monitors the position of the vessels and records the history of their courses. Vessels are represented as moving objects. Static objects represent fishing areas, harbors, etc.

We consider the following relations:

FishingShips(reference:string, name:string, voyages:movement)
ForbiddenAreas(name:string, geometry:polygon)
Harbors(name:string, location:polygon)

Notice that the entire movement of a vessel is represented as an attribute in the relation FishingShips.

We also need to define a predicate, denoted by $notEmpty()$, that receives as argument a movement value and returns $True$ if the value received is not empty; otherwise, the predicate returns $False$. Temporal values are in the format MM/DD/YYYY:hh:mm.ss. If time values hh, mm and ss, are omitted, it is assumed that they are equal to zero. Time intervals are represented by $\langle t_1, t_2 \rangle$ where $t_1$ and $t_2$ are two temporal values representing the bounds of the interval.

The queries correspond to the kinds of questions that IGP would like to be able to answer based on their database system. They will be expressed here using an SQL-like syntax.

**Q$_1$:** To start with, let's search for the activity of fishing ship "P01" since May 15, 2000.

```
SELECT   After(FishingShips.voyages, 05/15/2000)
FROM     FishingShips
WHERE    FishingShips.reference = "P01";
```

This query uses the SQL standard predicate = to select fishing ship "P01" and a movement operation $After$ to filter the desired voyages. The result is a movement ready to be used , for example, by a displaying tool.

**Q$_2$:** Let's now determine the location of the ships on May 31, 2000, 10:00AM.

```
SELECT   x.name,
         SpatialPathProjection(
             At(x.voyages, 05/31/2000:10:00.00))
FROM     FishingShips x;
```

This query returns a pair of values for each fishing ship, where the first one is the name of a fishing ship and the second is the location on May 31, 2000, 10:00AM. Each location is obtained using an operation *At* to filter the movement according to the time specified and then a spatial projection returns a point corresponding to the location of the fishing ship at that instant.

**$Q_3$:** Authorities want to investigate who was responsible for a spread of waste in the sea and want to know the behavior of all vessels that could have been in the area.

```
SELECT   x.name, PossiblyIn(x.voyages, :PollutedArea)
FROM     FishingShips x
WHERE    notEmpty(PossiblyIn(x.voyages, :PollutedArea));
```

The value *PollutedArea* is an user-defined polygon representing the area where the spread of waste has occurred. Predicate *notEmpty*() applied to operation *PossiblyIn* allows selecting only the fishing ships that could have been in that area. The query returns pairs of values, where the first is a name of a fishing ship and the second is the corresponding movement in that area.

**$Q_4$:** Authorities can only apply penalties when they are able to guarantee that a fishing ship has been in a forbidden area. The following query returns the name of the fishing ships that have been in the *Blue Coast* reservation.

```
SELECT   x.name
FROM     FishingShips x, ForbiddenAreas y
WHERE    y.name = "Blue Coast reservation"
AND      notEmpty(SurelyIn(x.voyages, y.geometry));
```

Operation *SurelyIn* returns the parts of the movements for which it is possible to assure that a fishing ship was inside the forbidden area. Applying the predicate *notEmpty*() to result of the previous operation allows selecting the tuples of the required fishing ships.

**$Q_5$:** Let's search for the fishing ships that were closer than 0.2 miles from vessel "P01" at May 28, 2000.

```
SELECT   x.name
FROM     FishingShips x, FishingShips y
WHERE    x.reference = "P01"
AND      y.reference ≠ x.reference
AND      notEmpty(ProbablyWithinDistance(
         During(x.voyages,⟨05/28/2000,05/29/2000⟩),
         During(y.voyages,⟨05/28/2000,05/29/2000⟩),0.2));
```

The first condition allows selecting the fishing ship with reference "P01". The second one avoids comparing the distance of "P01" with itself. Finally, the third condition selects the movement of "P01" and the movement of another fishing ship, during May 28, 2000, and returns the parts of the movement of "P01" for which the distance from the other movement is inferior to 0.2 miles. Predicate *notEmpty*() allows selecting only the tuples of relation $y$ conforming the specified relationship. The result is a set of candidates: there may be fishing ships selected that have not been closer to "P01" than the distance specified, and there may be fishing ships that have been closer than 0.2 miles from "P01" and have not been selected.

**$Q_6$:** The following query allows determining when "P01" has gone out the harbor "Sines" on May 31, 2000.

```
SELECT   TemporalProjection(SurelyLeaves(
            During(x.voyage, ⟨05/31/2000, 06/01/2000⟩),
            y.geometry))
FROM     FishingShips x, Harbors y
WHERE    x.reference = "P01"
AND      y.name = "Sines";
```

The two conditions allow selecting the fishing ship "P01" and the harbor "Sines". The *During* operation allows selecting the movement of P01 during May 31, 2000. *SurelyLeaves* operation compares the section of the movement selected against the geometry of the harbor and returns a new movement section. Finally, *TemporalProjection* extracts the time interval associated with the movement selected. Its is guaranteed that the vessel P01 leaved the harbor during that time interval.

**$Q_7$:** How many miles has "P01" covered during 1999.

```
SELECT   Lenght(SpatialProjection(
             During(x.voyage, ⟨01/01/1999, 01/01/2000⟩)))
FROM     FishingShips x
WHERE    x.reference = "P01";
```

The first step consists in selecting the movement of "P01" during 1999. Follows a spatial projection operation returning the line corresponding to the selected movement. Finally, we assume to exist a spatial operation, *Length*, that returns the length of the line corresponding to the distance covered by "P01" in 1999.

**$Q_8$:** Let's now check which are the forbidden areas having the greatest number of infractions.

```
SELECT     y.name, Sum(Elapsed(TemporalProjection(
                ProbablyIn(x.voyages, y.geometry))))
FROM       FishingShips x, ForbiddenAreas y
GROUP BY   y.name
ORDER BY   2;
```

Operation *ProbablyIn* allows selecting for each fishing ship, the sections of movement where it could have been within a forbidden area. Applying a temporal projection to the selected movements, allows obtaining the time intervals when the fishing ships were within a forbidden area. *Elapsed*, is a temporal operation, that we assume to exist, returning the amount of time represented in a time interval. The *Group By* clause engenders the aggregation of the returned values accordingly to the name of the forbidden areas. The result is a set of pairs, composed by the name of a restricted area and a value, such that, value is the sum of amounts of time for which fishing ships probably were in that forbidden area. The result is ordered by the numeric value.

## 5. RELATED WORK

Recently, strongly encouraged by the ChoroChronos project, the temporal and spatial databases communities started with systematic research on data models and languages for the representation of spatio-temporal data. There are two main approaches. One is based on the constraint databases paradigm and consists of a non-specialized abstract data model and a query language [6]. The model is based on linear constraints and allows representing data of arbitrary dimensions. Operations are basically those of relational algebra on infinite point sets. The other follows an abstract data types approach [7, 8]. The design involves base and spatial types, as well as temporal versions of these. The idea is the definition of an algebra over non-temporal data types that are then lifted to operations over temporal types. The integration of lifted operations into SQL has also been analyzed [3] and the concept of sliced representation, allowing to represent discretely a temporal development within a units data structure, is introduced in [5].

[4] introduces a framework based on an extension of an object-oriented database system for the development of spatio-

temporal applications. Spatial, temporal and spatio-temporal operations for a spatio-temporal query language are informally introduced.

Multimedia applications are also interested in handling the spatial and temporal relationships of data such as image, graphics, video or animation. [14] defines a model for the representation of spatio-temporal relationships between objects in multimedia applications. A model that allows information retrieval from a multimedia database system, based not only on topological relationships between static spatial objects, but also on temporal relationships between moving objects in a sequence is proposed in [10].

The works mentioned above use different approaches to combine space and time at an abstract level. However, none of them takes into account the integration of spatio-temporal operations and uncertainty, to cope with the characteristics of real-world observation-driven systems, as proposed in this paper.

The representation of uncertainty in the position of moving objects is analyzed in [12]. Locations for objects between observations are determined by applying an interpolation method. Imprecision introduced by the interpolation method, as well as the errors introduced by measurement instruments, are also analyzed, and a technique is proposed for the quantification of such errors.

## 6. CONCLUSIONS AND FUTURE WORK

In this paper we analyze the representation of movement in database systems. We consider the case of observations-based systems, monitoring the behavior of moving objects represented as points and having trajectories with few restrictions in movement. Uncertainty is an intrinsic feature of such movement's representation. The position of a moving object at an arbitrary time can however be determined within known precision. Using this, we propose a new set of spatio-temporal operations with semantics that embed uncertainty. These operations are suitable to be integrated in a moving objects query language. The main contribution of this work lies in the combination of spatio-temporal operations with spatio-temporal uncertainty.

This work fits in a research framework concerned with spatio-temporal representations for moving points and covering the data model, query language, physical representation and benchmarking. The definition of an object-oriented logical data model is being completed. The associated query language is intended to incorporate the set of operations proposed here. As for the physical representation, we have already defined a model of movement representation, based on linear approximations [9]. There is also preliminary work on access method for the efficient support of movement related operations. The benchmarking methodology for testing prototype applications has also been defined [13].

## 7. REFERENCES

[1] E. Clementini, P. Felice, and P. Oosterom. A small set of formal topological relationships suitable for end-user interaction. In *Proceedings of the Third International Symposium on Large Spatial Databases, SSD '93*, volume 692 of *Lecture Notes in Computer Science*, pages 277–295. Springer, 1993.

[2] M. J. Egenhofer. Spatial relations: Models and inferences (tutorial). *5th International Symposium on spatial databases*, 1999.

[3] M. Erwig and M. Schneider. Developments in spatio-temporal query languages. In *Proceedings of the 10th International Workshop on Database and Expert Systems Applications*, pages 441–449. IEEE Computer Society, 1999.

[4] G. Faria, C. Medeiros, and M. Nascimento. An extensible framework for spatio-temporal database applications. In *Proceedings of the 10th International Conference on Scientific and Statistical Database Management*, pages 202–205. IEEE Computer Society, 1998.

[5] L. Forlizzi, R. Guting, E. Nardelli, and M. Schneider. A data model and data structures for moving objects databases. In *Proceedings of the 2000 ACM SIGMOD Conference on Management of data*, pages 319–330, 2000.

[6] S. Grumbach, P. Rigaux, and L. Segoufin. Spatio-temporal data handling with constraints. In *Proceedings of the 6th International Symposium on Advances in Geographic Information Systems (ACMGIS-98)*, pages 106–111. ACM Press, 1998.

[7] R. Guting, M. Bohlen, M. Erwig, C. Jensen, N. Lorentzos, M. Schneider, and M. Vazirgiannis. A foundation for representing and querying moving objects. *To appear in ACM Transactions on Database Systems.*

[8] M. S. M. Erwig, R. Guting and M. Vazirgiannis. Spatio-temporal data types: An approach to modeling and querying moving objects in databases. *GeoInformatica*, 3(3):269–296, 1999.

[9] J. Moreira, J.-M. Saglio, and C. Ribeiro. Representation and manipulation of moving points: An extended data model for location estimation. *Journal of Cartography and Geographic Information Systems (CaGIS), ACSM*, 26(2), 1999.

[10] M. Nabil. *Modelling and retrieving multimedia data using spatial and temporal information.* PhD thesis, University of New South Wales, Sydney, Australia, 1997.

[11] D. Papadias, Y. Theodoridis, and T. Sellis. The retrieval of direction relations using R-trees. In *Proceedings of the 5th International Conference on Database and Expert Systems Applications*, volume 856, pages 173–182. Lecture Notes in Computer Science, 1994.

[12] D. Pfoser and C. Jensen. Capturing the uncertainty of moving-object representations. *Lecture Notes in Computer Science*, 1651:111–132, 1999.

[13] J.-M. Saglio and J. Moreira. Oporto: A realistic scenario generator for moving objects. In *Proceedings of the 10th International Workshop on Database and Expert Systems Applications*, pages 426–432. IEEE Computer Society, 1999.

[14] M. Vazirgiannis, Y. Theodoridis, and T. Sellis. Spatio-temporal composition in multimedia applications. In *Proceedings of IEEE-ICSE 96, Workshop on Multimedia Software Development.* IEEE Computer Society, 1996.

[15] T.-S. Yeh and B. Cambray. Modelling highly variable spatio-temporal data. *Sixth Australian Database Conference (ADC95)*, 1995.