Aggregation Languages for Moving Object and Places of Interest

Leticia I. Gómez
Instituto Tecnólogico de de Buenos Aires
Av. E. Madero 399
Bs.As., Argentina
lgomez@itba.edu.ar

Bart Kuijpers
University of Hasselt,
Departement WNI
Gebouw D, B-3590
Diepenbeek, Belgium
bart.kuijpers@uhasselt.be

Alejandro A. Vaisman
Universidad de Buenos Aires
Ciudad Universitaria PI,
Bs.As., Argentina
avaisman@dc.uba.ar

ABSTRACT

We address *aggregate queries* over GIS data and moving object data, where non-spatial information is stored in a data warehouse. We propose a formal data model and query language to express complex aggregate queries. Next, we study the compression of trajectory data, produced by moving objects, using the notions of *stops and moves*. We show that stops and moves are expressible in our query language and we consider a fragment of this language, consisting of regular expressions to talk about temporally ordered sequences of stops and moves. This fragment can be used not only for querying, but also for expressing data mining and pattern matching tasks over trajectory data.

Categories and Subject Descriptors

H.2.8 [**Database Applications**]: Spatial Databases and GIS; H.4.2 [**Information Systems Applications**]: Decision Support

General Terms

Design, Languages

Keywords

GIS, OLAP, View Materialization

1. INTRODUCTION

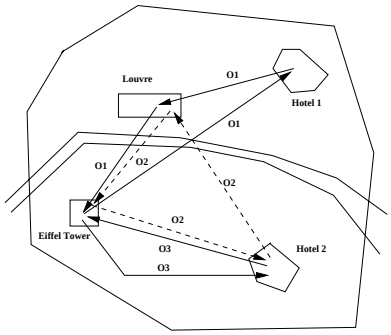
Geographic Information Systems (GIS) have been extensively used in various application domains, ranging from economical, ecological and demographic analysis, to city and route planning [14]. In recent years, *time* is playing an increasingly important role in GIS and spatial data management [11]. One particular line of research in this direction, introduced by Wolfson [15, 16, 17], concerns *moving object data*. Moving objects, carrying location-aware devices, produce trajectory data in the form of a sample of (O_{id}, t, x, y)-tuples, that contain object identifier and time-space information. Recently, the notions of *stops* and *moves* were introduced [2, 10]. These concepts serve to compress the trajectory data that is produced by moving objects using application-dependent places of interest. A designer may want to select a set of places of interest that are relevant to her application. For instance, in a tourist application, such places can be hotels, museums and churches. In a traffic control application, they may be road segments, traffic lights and junctions, stored in GIS layers. If a moving object spends a sufficient amount of time in a place of interest, this place is considered a stop of the object's trajectory. In between stops, the trajectory has moves. Thus, we can replace a raw trajectory given by (O_{id}, t, x, y)-tuples by a sequence of application-relevant stops and moves. This also adds semantic information valuable for querying and analysis.

In this paper, we are interested in *aggregate queries* over GIS data and moving object data. Typically, when aggregation becomes important, it is advisable to organize the non-spatial data in a GIS in a data warehouse. In a data warehouse, numerical data are stored in fact tables built along several dimensions. For instance, we may store the sales amounts in a fact table over three dimensions store, time and product. In general, dimensions are organized into aggregation hierarchies. For example, stores can aggregate over cities which in turn can aggregate into regions and countries. Each of these aggregation levels can also hold descriptive attributes like city population, the area of a region, etc. On Line Analytical Processing(OLAP) [6] provides tools and algorithms that allow efficiently querying a data warehouse.

1.1 Motivation and Running Example

We motivate our work with the following example. Figure 1 (left) shows a simplified map of Paris, containing two hotels, denoted Hotel 1 and Hotel 2 (H1 and H2 from here on), the Louvre and the Eiffel tower. We consider three moving objects, O1, O2 and O3. Object O1 goes from H1 to the Louvre, the Eiffel tower, spends just a few minutes there, and returns to the hotel. Object O2 goes from H2 to the Louvre, the Eiffel tower, (spending a couple of hours visiting each place), and returns to the hotel. Object O3 leaves H2 to the Eiffel tower, visits the place, and returns to H2. An example of these trajectory samples is shown on the right hand side of Figure 1. All points of the same trajectory are temporally ordered (and stored together). In what follows,

SAC'08 March 16-20, 2008, Fortaleza, Ceará, Brazil



O_{id}	t	x	y
O_1	1	x_1	y_1
O_1	2	x_2	y_2
O_1	3	x_3	y_3
O_1	4	x_4	y_4
...	...	...	...
O_2	5	x_5	y_5
O_2	6	x_6	y_6
O_2	7	x_7	y_7
...	...	...	...
O_3	4	x_5	y_5
O_3	5	x_8	y_8
O_3	6	x_9	y_9
...	...	...	...

Figure 1: Running example

we will use the object identifier as the trajectory identifier, unless specified.

Many useful applications open in this scenario. For instance, a GIS user may be interested in finding out trajectory information, like "number of persons going from H1 to the Louvre and then to the Eiffel tower (stopping for visiting both places) in the same day", or "number of persons going from a hotel in the left bank of the Seine, to the Louvre in the mornings during the last summer season". An analyst may also want to discover hidden information using data mining techniques. For instance, she would like to identify interesting patterns in the trajectory data using association rule mining. Complex queries that aggregate non-spatial information, and also involve GIS and moving object data, must also be addressed. For instance, "total sales in museums such that people visit them before going to the Eiffel Tower in the same day, and that are located in the left bank of the Seine".

1.2 Related Work

The field of moving objects databases has been extensively studied in the last ten years, specially regarding data modeling an indexing. Güting and Schneider [4] provide a good reference to this large corpus of work. Wolfson [17] *et al* stated a set of capabilities that a moving object database must have, and introduced the DOMINO system. Hornsby and Egenhofer [5] introduced a framework for modeling moving objects, that supports viewing objects at different granularities, depending on the sampling time interval. The basic modeling element they consider is a *geospatial lifeline,* which is composed of triples of the form $< Id, location, time >$, where Id is the identifier of the object, *location* is given by x-y coordinates, and *time* is the timestamp of the observation. The possible positions of an object between two observations is estimated to be within two inverted half-cones that conform a *lifeline bead*, whose projection over the x-y plane is an ellipse. For mining trajectories in road networks, Brakatsoulas *et al* [1] proposed to enrich trajectories of moving objects with information about the relationships between trajectories (e.g., *intersect*, *meets*), and between a trajectory and the GIS environment (s*tay within*, *bypass*, *leave*). They also propose a mining language denoted SML (for Spatial Mining Language). However, the language does not take advantage of the particular characteristics that moving object data present. Lee *et al.* [8] aim at discovering common sub-trajectories, and also use the a partitioning strategy, proposing a partition-and-group framework for clustering trajectories.

Techniques that add semantic information to trajectory data have been recently proposed. Mouza and Rigaux [10] presented a model where trajectories are represented by a sequence of moves. They propose a query language based on regular expressions, aimed at obtaining so-called mobility patterns. Note that this language, as well as the proposals commented above, does not relate trajectories with the GIS environment. With a similar idea, Damiani *et al.* [2] introduced the concept of stops and moves, in order to enrich trajectories with semantically annotated data.

Data aggregation is still an open field, either in GIS or in a moving objects scenario. Meratnia and de By [9] have tackled the topic of aggregation of trajectories, identifying similar trajectories and merging them in a single one, by dividing the area of study into homogeneous *spatial units*. Papadias *et al* [12] index historical aggregate information about moving objects. Kuijpers *et al.* [7] proposed a taxonomy of aggregation queries on moving object data.

1.3 Contributions and Paper Organization

Our proposal studies problems not addressed in the efforts commented above. To begin with, we present a conceptual model and aggregation language that integrates with GIS and non-spatial data (stored in the data warehouse) in a unified framework (Section 2). At the basis of this aggregation query language is a multi-sorted first-order query language $\mathcal{L}_{mo}$ for moving object and GIS data in which one can specify properties of moving objects, geometric elements of GIS layers and OLAP data storing the non-spatial GIS data (Section 3). We give a geometric definition of stops and moves and show that they are computable and study the compression of trajectory data, using the notions of stops and moves (see Section 4). Finally, we sketch a sub-language of $\mathcal{L}_{mo}$ that allows us to talk about temporally ordered sequences of stops and moves. The syntax of this language is given in the form of regular expressions (see Section 5). We show that this language considerably extends the language proposed by Mouza and Rigaux [10], and can be used to express data mining and pattern matching tasks over trajectory data.

2. DATA MODEL

Spatial Data. Our model for spatio-temporal data, builds on a previously introduced one [3], which supports just spatial data. Here we give an overview, in order to help the understanding of the remainder of the presentation. Figure 2 (left) depicts the schema of a so-called *GIS dimension*, basically a set of hierarchies (one for each GIS layer). The bottom level of each hierarchy, denoted the *Algebraic part* of the dimension, contains the infinite points in a layer, and could be described by means of linear algebraic equalities and inequalities [13]. Above this part there is the *Geometric part,* that stores the identifiers of the geometric elements of GIS, and it is used to solve the geometric part of a query (i.e. find the polylines in a river representation). Each point in the Algebraic part may correspond to one or more elements in the Geometric part. Thus, at the *GIS dimension instance* level we will have rollup *relations* (denoted $r_L^{geom_1 \to geom_2}$). These relations map, for example, points in the Algebraic part, to geometry identifiers in the Geometric part. For example, $r_{L_{province}}^{point \to Pg}(x, y, pg_1)$ says that point (x, y) corresponds to a polygon identified by pg_1 in the Geometric part, in the layer

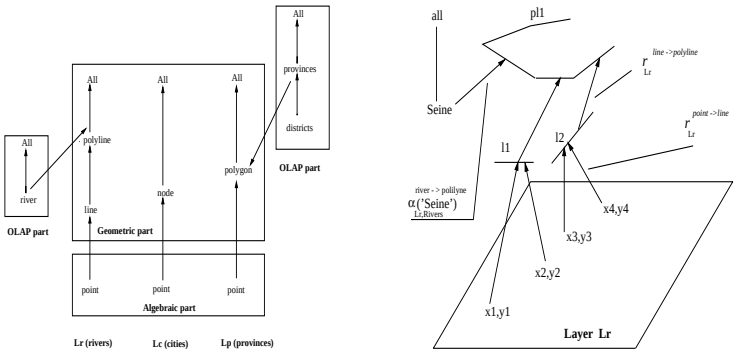


Figure 2: A GIS dimension schema (left) and A GIS dimension instance (right)

representing provinces (note that a point may correspond to more than one polygon, o to more than one polylines that intersect with each other).

Finally, there is the *OLAP part* of the dimension. This part contains the conventional OLAP structures. The levels in the geometric part are associated to the OLAP part via a function, denoted $\alpha_{L,D}^{dimLevel \to geom}$. For instance, $\alpha_{L_r,Rivers}^{riverId \to g_r}$ associates information about a river in the OLAP part (the $riverId$), to the identifier of a polyline (g_r) in a layer containing rivers (L_r) in the Geometric part.

Example 1. *Figure 2 (left) shows the schema of a GIS dimension, where we have defined three layers, for rivers, cities, and provinces, respectively. The schema is composed of three graphs; the graph for rivers contains edges saying that a point (x, y) in the algebraic part relates to a line identifier in the geometric part, and that in the same portion of the dimension, this line aggregates on a polyline identifier.*

In the OLAP part we have information given by two dimensions, representing districts and rivers, associated to the corresponding graphs, as the figure shows. For example, a river identifier at the bottom layer of the Rivers *dimension in the OLAP part, is mapped to the polyline dimension level in the geometric part in the graph in the rivers layer L_r.*

Figure 2 (right) shows a portion of a GIS dimension instance of the rivers layer L_r in the dimension schema of the schema in the left of the figure. Here, an instance of a GIS dimension in the OLAP part is associated to the polyline pl_1, which corresponds to the Seine river. For simplicity we only show four different points at the point level $\{(x_1, y_1), \ldots, (x_4, y_4)\}$. There is a relation $r_{L_r}^{point \to line}$ containing the association of points to the lines in the line *level, and a relation $r_{L_r}^{line \to polyline}$, between the line and polyline levels, in the same layer.* □

Elements in the geometric part can be associated with *facts*, each fact being quantified by one or more *measures*. Besides, there may also be classical OLAP fact tables.

Moving Object Data Representation. Besides the static information representing geometric components, time in the OLAP part will be represented by a *Time* dimension (actually, there could be more than one Time dimension, supporting different notions of time). Moving objects are integrated in the former framework using a distinguished fact table denoted *Moving Object Fact Table*. First, we say what a trajectory is.

Definition 1 (Trajectory). *A* trajectory *is a list of time-space points $\langle (t_0, x_0, y_0), (t_1, x_1, y_1), ..., (t_N, x_N, y_N) \rangle$, where $t_i, x_i, y_i \in \mathbb{R}$ for $i = 0, ..., N$ and $t_0 < t_1 < \cdots < t_N$. We call the interval $[t_0, t_N]$ the* time domain *of the trajectory. For the sake of finite representability, we may assume that the time-space points (t_i, x_i, y_i), have rational coordinates.* □

Definition 2 (Moving Object Fact Table). *Given a finite set $\mathcal{T}$ of trajectories, a* Moving Object Fact Table *(MOFT) for $\mathcal{T}$ is a relation with schema $< Oid, T, X, Y >$, where Oid is the identifier of the moving object, T represents time instants, and X and Y represent the spatial coordinates of the objects. An instance $\mathcal{M}$ of the above schema contains a finite number of tuples of the form (O_i, t, x, y), that represent the position (x, y) of the object O_i at instant t, for the trajectories in $\mathcal{T}$. The table in the Figure 1(right),is the MOFT for our running example.* □

3. QUERY LANGUAGE

The aggregation queries we address in this paper are based on a first-order moving object query language we denote $\mathcal{L}_{mo}$, and they are of the following types: (1) the Count operator applied to sets of the form $\{O_{id} \mid \phi(O_{id})\}$, where moving objects identifiers satisfying some $\mathcal{L}_{mo}$-definable property ϕ are collected; (2) the Count operator applied to sets of the form $\{(O_{id}, t) \mid \phi(O_{id}, t)\}$, where moving objects identifiers combined with time moments, satisfying some $\mathcal{L}_{mo}$-definable property ϕ, are collected (assuming that this set is finite; otherwise the count is undefined); (3) the Count operator applied to sets of the form $\{(O_{id}, t, x, y) \mid \phi(O_{id}, t, x, y)\}$, where moving objects id's combined with time and space coordinates, satisfying some $\mathcal{L}_{mo}$-definable property ϕ, are collected (assuming that this set is finite); (4) the Area operator applied to sets of the form $\{(x, y) \in \mathbb{R}^2 \mid \phi(x, y)\}$, which define some $\mathcal{L}_{mo}$-definable part of the plane $\mathbb{R}^2$ (assuming that this set is linear and bounded); (5) the Count, Max and Min operators applied to sets of the form $\{t \in \mathbb{R} \mid \phi(t)\}$, when the $\mathcal{L}_{mo}$-definable condition ϕ defines a finite set of time instants ; (6) the Max-l, Min-l, Avg-l and TimeSpan-l operators applied to sets of the form $\{(t_s, t_f) \in \mathbb{R}^2 \mid \phi(t_s, t_f)\}$, which represents an $\mathcal{L}_{mo}$-definable set of time intervals; (7) the Area operator applied to sets of the form $\{g_{id} \mid \phi(g_{id})\}$, where identifiers of elements of some geometry (in the geometric part of our data model), satisfying an $\mathcal{L}_{mo}$-definable ϕ are collected; (8) the Count operator applied to sets of the form $\{(O_i d, g_{id}, t_s, t_f) \mid \phi(O_{id}, g_{id}, t_s, t_f)\}$. Obviously, the above list is not complete, but it covers the most interesting and usual cases.

To complete the description of our moving-object aggregation language, the query language $\mathcal{L}_{mo}$ remains to be defined. In the $\mathcal{L}_{mo}$-definable sets considered above, we can see that there are variables of different kinds, like O_{id}, t, x, y and g_{id}. In fact, $\mathcal{L}_{mo}$ is a multi-sorted first-order logic using variables of these types to define sets as considered above.

Definition 3. *The first-order query language $\mathcal{L}_{mo}$ has four types of variables:* real variables $x, y, t, \ldots$; name variables $O_{id}, ...$; geometric identifier variables $g_{id}, ...$ *and* dimension level variables $a, b, c, ...$, *(which are also use for dimension level attributes). Besides (existential and universal) quantification over all these variables, and the usual logical connectives $\wedge, \vee, \neg ...$, we consider the following functions and relations to build atomic formulas in $\mathcal{L}_{mo}$: (a) for*

every rollup function in the OLAP part, we have a function symbol $f_{D_k}^{G_i \to G_j}$, where G_i and G_j are geometries and D_k is a dimension; (b) analogously, for every rollup relation in the GIS part, we have a relation symbol $r_{L_k}^{G_i \to G_j}$, where G_i and G_j are geometries and L_k is a layer; (c) for every α relation associating the OLAP and GIS parts in some layer L_i, we have a relation symbol $\alpha_{L_k,D_\ell}^{A_i \to G_j}$, where A_i is a OLAP dimension level and G_j is a geometry, L_k is a layer and D_ℓ is a dimension; (d) for every dimension level A*, and every attribute B of A, denoted $A.B$, there is a function $\beta_{D_k}^{A \to B}$ that maps elements of A to elements of B in dimension D_k; (e) we have functions, relations and constants that can be applied to the alpha-numeric data in the OLAP part (e.g., we have the $\in$ relation to say that an element belongs to a dimension level, we may have $<$ on income values and the function concat on string values); (f) for every MOFT, we have a 4-ary relation $\mathcal{M}_i$; (g) we have arithmetic operations $+$ and $\times$, the constants* 0 *and* 1*, and the relation $<$ for real numbers. (h) finally, we assume the equality relation for all types of variables. If needed, we may also assume other constants.* □

Definition 3 describes the syntax of the language $\mathcal{L}_{mo}$. The interpretation of all variables, functions, relation, and constants is standard, as well as that of the logical connectives and quantifiers. We illustrate the semantics through an elaborated example.

EXAMPLE 2. *Let us consider the query* "Total number of buses per hour running in the morning in the Paris districts with a monthly income of less than €1500,00."

We use the MOFT $\mathcal{M}$ (Figure 1, center), that contains the moving objects samples. For clarity, we will denote the geometry polygons by Pg, polylines by Pl and point by Pt. We use distr *to denote the level district in the OLAP part of the dimension schema. The GIS layer which contains district information is called L_d. The query returning the* region with the required income *is expressed:*
$\{(x,y) \mid \exists n \exists g_1 (r_{L_d}^{Pt \to Pg}(x,y,g_1) \ \wedge \ \alpha_{L_d,Distr}^{distr \to Pg}(n) = g_1 \ \wedge \ \beta_{Distr}^{distr \to income}(n) < 1.500)\}$

In this expression, $r_{L_d}^{Pt \to Pg}(x,y,g_1)$ relates points to polygons in the district layer; the function $\alpha_{L_d,Distr}^{distr \to Pg}(n) = g_1$ maps the district identifier n in the OLAP part to the geometry identifier g_1 in the layer L_d (here Distr is the dimension in the OLAP part representing districts); and $\beta_{Distr}^{distr \to income}(n)$ maps the district identifier n to the value of the income *attribute which then is compared by an OLAP relation $<$ with an OLAP constant* 1.500*. The instants corresponding to the morning hours mentioned in the fact tables are obtained through the rollup functions in the Time dimension. We assume in the Time dimension a category denoted* timeOfDay*, rolling up to the dimension category* hour *(i.e., $timeOfDay \to hour$). The aggregation of the values in the fact table corresponding only to morning hours is computed with the following expression: $\mathcal{M}_{morning} = \{(Oid,t,x,y) \mid f_{Time}^{hour \to timeOfDay}(t) =$ "Morning" $\wedge \mathcal{M}(Oid,t,x,y)\}$. In this formula "Morning" appears as a constant in the OLAP part. Finally, the query we discuss reads:*
$\text{COUNT}\{(Oid,t) \mid (\exists x)(\exists y)(\exists g)(\exists g_1)(\exists n) \ (r_{L_d}^{Pt \to Pg}(x,y,g_1) \ \wedge \ \mathcal{M}_{morning}(Oid,t,x,y) \ \wedge \ \alpha_{L_d,Distr}^{distr \to Pg}(n) = g \ \wedge \ \beta_{Distr}^{distr \to income}(n) < 1,500)\}$. □

PROPOSITION 1. *Moving object queries expressible in $\mathcal{L}_{mo}$ are computable. The proposed aggregation operators are also computable.*

The proof follows from the restrictions imposed on the applicability of aggregation operators make sure that they can be effectively evaluated.

4. THE STOPS AND MOVES FACT TABLE

In a GIS scenario, stops and moves depend on the *places of interest* of a particular application. For instance, in a tourist application, places of interest may be hotels, museums and churches. In a traffic application, places of interest may be road segments, road junctions and traffic lights. First, we define the notion of "places of interest of an application".

DEFINITION 4. *[Places of Interest] A* place of interest (PoI) *C is a tuple (R_C, Δ_C), where R_C is a (topologically closed) polygon, polyline or point in $\mathbb{R}^2$ and Δ_C is a strictly positive real number. The set R_C is called the* geometry *of C and Δ_C is called its* minimum duration*. The* places of interest of an application *$\mathcal{P}_\mathcal{A}$ is a finite collection of PoIs with mutually disjoint geometries.* □

DEFINITION 5. *[Stops and moves of a trajectory] Let $T = \langle (t_0,x_0,y_0),(t_1,x_1,y_1),...,(t_n,x_n,y_n)\rangle$ be a trajectory. Also, $\mathcal{P}_\mathcal{A} = \{C_1 = (R_{C_1},\Delta_{C_1}),...,C_N = (R_{C_N},\Delta_{C_N})\}$.*

A stop of T with respect to $\mathcal{P}_\mathcal{A}$ *is a maximal contiguous subtrajectory $\langle (t_i,x_i,y_i),(t_{i+1},x_{i+1},y_{i+1}),...,(t_{i+\ell},x_{i+\ell},y_{i+\ell})\rangle$ of T such that for some $k \in \{1,...,N\}$ the following holds: (a) $(x_{i+j},y_{i+j}) \in R_{C_k}$ for $j = 0,1,...,\ell$; (b) $t_{i+\ell} - t_i > \Delta_{C_k}$.*

A move of T with respect to $\mathcal{P}_\mathcal{A}$ *is: (a) a maximal contiguous subtrajectory of T in between two temporally consecutive stops of T; (b) maximal contiguous subtrajectory of T in between the starting point of T and the first stop of T; (c) a maximal contiguous subtrajectory of T in between the last stop of T and ending point of T; (d) the trajectory T itself, if T has no stops.* □

Intuitively, if a trajectory is inside a place of interest longer than the minimum duration Δ_C, the place of interest is a *stop*. We remark that our definition of stops and moves of a trajectory is arbitrary and can be modified in many ways. For example, if we would work with linear interpolation of trajectory samples, rather than with samples, the trajectory may leave a place of interest, not in a sample point, but in the interpolation. We could incorporate a tolerance for this kind of small exits in the definition.

PROPOSITION 2. *There is an algorithm that returns, for any input $(\mathcal{P}_\mathcal{A},T)$ with $\mathcal{P}_\mathcal{A}$ the places of interest of an application, and T a trajectory the stops of T with respect to $\mathcal{P}_\mathcal{A}$. This algorithm works in time $\mathcal{O}(n \cdot p)$, where p is the complexity of answering the point-query [14].* □

Note that a MOFT only provides the position of objects at a given instant. Sometimes we are not interested in such level of detail, but we look for more aggregated information instead. For example, we may want to know the how many people go from a hotel to a museum on weekdays. Or, we can even want to perform data mining tasks like inferring trajectory patterns that are hidden in the MOFT. These tasks require semantic information, not present in the MOFT. In

the best case, obtaining this information from that table will be expensive, because it would imply a join between this table and the spatial data. As a solution, we propose to use the notion of *Stops and Moves* in order to obtain a more concise MOFT, that can represent the trajectory in terms of places of interest, characterized as *Stops*. This table cannot replace the information provided by the MOFT, but complement it, allowing to quickly obtain information of interest without accessing the complete data set. In this sense, this concise MOFT, which we will denote SM-MOFT behaves like a summarized materialized view of the MOFT. The SM-MOFT will contain the object identifier, the identifier of the geometries representing the Stops, and the interval $[t_s, t_f]$ of the stop duration. Notice that we do not need to store the information about the moves, which remains implicit, because we know that between two stops there could only be a move or a transition. Also, if a trajectory passes through a PoI, but remains there an insufficient amount of time for considering the place a trajectory stop, the stop is not recorded in the SM-MOFT. Thus, for this application, two trajectories apparently identical, will be distinguished by these representation scheme.

DEFINITION 6 (SM-MOFT). *Let $\mathcal{P}_\mathcal{A} = \{C_1 = (R_{C_1}, \Delta_{C_1}), ..., C_N = (R_{C_N}, \Delta_{C_N})\}$ be the set of PoI of an application, and let $\mathcal{M}$ be a MOFT. The* SM-MOFT $\mathcal{M}^{sm}$ of $\mathcal{M}$ with respect to $\mathcal{P}_\mathcal{A}$ *consist of the tuples $(O_{id}, g_{id}, t_s, t_f)$ such that (a) O_{id} is the identifier of a trajectory in $\mathcal{M}$; (b) g_{id} is the identifier of the geometry of a PoI $C_k = (R_{C_k}, \Delta_{C_k})$ of $\mathcal{P}$ such that the trajectory with identifier O_{id} in $\mathcal{M}$ has a stop in this PoI during the time interval $[t_s, t_f]$. This interval is called the* stop interval *of this stop.* □

PROPOSITION 3. *There is an $\mathcal{L}_{mo}$ formula $\phi_{sm}(O_{id}, g_{id}, t_s, t_f)$ that defines the SM-MOFT $\mathcal{M}^{sm}$ of $\mathcal{M}$ with respect to $\mathcal{P}_\mathcal{A}$.* □

We omit the proof of this property but remark that the use of the formula $\phi_{sm}(O_{id}, g_{id}, t_s, t_f)$ allows us to speak about stops and moves of trajectories in $\mathcal{L}_{mo}$. We can therefore add predicates to define stops and moves of trajectories as syntactic sugar to $\mathcal{L}_{mo}$.

5. A LANGUAGE FOR STOPS AND MOVES

We will show how the language $\mathcal{L}_{mo}$ yield sub-languages that can address many interesting aggregation queries for moving objets in a GIS environment. We will sketch a query language based on regular expressions, along the lines proposed by Mouza and Rigaux [10]. However, our language goes far beyond, taking advantage of the integration between GIS, OLAP and moving objects provided by the data model. Moreover, queries that do not require access to the MOFT can be evaluated efficiently using the SM-MOFT, $\mathcal{M}^{sm}$. The idea is based on the construction of the $\mathcal{SM}$-Graph defined below.

We will assume that there will be a different dimension for each type of (application-dependant) place of interest in the OLAP part of the model. For instance, there will be a dimension for hotels, with bottom level *hotelId*, or a dimension for restaurants, with bottom level *restaurantId*. Aggregation levels can be defined as required. There will also be a layer in the *Geometric part* of the GIS dimension, that could be designed in different ways. For simplicity, we

Oid	G_{id}	t_s	t_f
O_1	H_1	0	10
O_1	M_1	15	30
O_1	M_2	40	50
O_1	M_2	60	70
O_1	M_3	80	100
O_2	T_1	120	150
O_2	T_2	180	200
O_3	H_1	220	240
O_3	T_2	280	340
O_3	M_3	410	*Now*

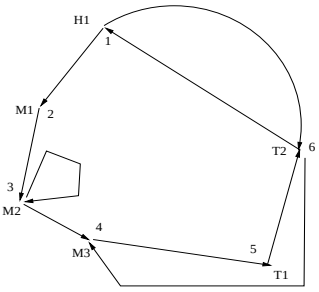


Figure 3: An SM-MOFT (left), and its $\mathcal{SM}$-Graph (right)

assume that all places of interest with the same geometry are stored together. For instance, there will be a layer (i.e., a hierarchy graph) for polygons, and/or one hierarchy for lines. There are also the functions introduced in Section 2. For example, $\alpha^{hotelId \to Pg}_{Lp,Hotel}$ maps a hotel identifier to a polygon representing it, in a layer for polygonal PoIs.

DEFINITION 7 ($\mathcal{SM}$-GRAPH). *Let us consider a trajectory sample $\mathbb{T}$ of moving objects, the PoIs of an application $\mathcal{P}_\mathcal{A} = \{C_1 = (R_{C_1}, \Delta_{C_1}), ..., C_N = (R_{C_N}, \Delta_{C_N})\}$, a MOFT $\mathcal{M}$, and its SM-MOFT $\mathcal{M}^{sm}$ with respect to A. Also, for clarity but w.l.o.g., consider that all the tuples in $\mathcal{M}^{sm}$ are* ordered *according to their stop interval attributes, that is, if t_1 and t_2 are two consecutive tuples in $\mathcal{M}^{sm}$, $t_1.t_s < t_1.t_f < t_2.t_s < t_2.t_f$. An $\mathcal{SM}$-Graph for $\mathcal{M}^{sm}$, denoted $\mathcal{G}(M^{sm})$, is a graph constructed as follows: (1) For each $g_{id} \in \prod_{G_{id}}(\mathcal{M}^{sm})$ there is a node v in $\mathcal{G}$, denoted $v(g_{id})$, with a* node number $n \in \mathbb{N}$, *different for each node. (2) There is an edge m in $\mathcal{G}$ between two nodes $v(g_{id_1})$ and $v(g_{id_2})$, for every pair of t_1, t_2 of consecutive tuples in $\mathcal{M}^{sm}$ with the same O_{id}. (3) For each node $v \in \mathcal{G}$ the* extension *of v, denoted* ext(v) *is given by the identifier of the PoI that represents the node in the OLAP part of the model. (4) For each node $v \in \mathcal{G}$ the* label *of v, denoted* label(v) *is the name of the dimension of the PoI in the OLAP Part (i.e., the name of the dimension D mentioned above). (5) For each node $v \in \mathcal{G}$ the* stop temporal elements *of v, denoted* STE(v) *is a set of stop intervals $\{I_1, ..., I_k\}$ (technically, a temporal element), such that there is an interval $I_i \in$* STE(v) *for each edge incoming to v in $\mathcal{G}$.*

Figure 3 (left) shows an SM-MOFT table for one moving object's trajectory. The distinguished term "Now" indicates, as usual in temporal databases, the *current* time. We denote H_i, M_i, and T_i, hotels, museums and tourist attractions, respectively. Figure 3 (right) shows the corresponding $\mathcal{SM}$-Graph. As an example, the extension of node 3 is $ext(3) = M2$, its label is $label(3) = Museums$, and $STE(3) = \{[80, 100], [410, Now]\}$.

Now we are ready to define our query language based on Stops and Moves. The language combines in a single expression, the notion of regular expressions and first order constraints. The $\mathcal{SM}$-Graph $\mathcal{G}$ can be seen as an automaton accepting regular expressions over the places of interest.

DEFINITION 8 (R.E. FOR STOPS AND MOVES). *A* regular expression on stops and moves, *denoted* smRE *is an expression generated by the grammar*

$$E \longleftarrow dim \mid dim[cond] \mid (E)^* \mid E.E \mid \epsilon \mid ?$$

where dim ∈ D (a set of dimension names in the OLAP part), ϵ is the symbol representing the empty expression, "." means concatenation, and cond represents a condition that can be expressed in $\mathcal{L}_{mo}$. The term "?" is a wildcard meaning "any sequence of any number of dim". □

Aggregation is built on top of *smRE*: for each trajectory T in an SM-MOFT such that there is a sub-trajectory of T that matches the *smRE*, the query returns the O_{id} of T. The following examples provide an overview of the language. We begin with the query "*Total number of trajectories from a Hilton hotel to a tourist attraction, stopping at a museum*," which reads in *smRE*:
COUNT(H[name = "Hilton"].?.M.?.T)

As another example, the query "*Total number of trajectories that went from a Hilton hotel to the Louvre, in the morning*" is expressed in *smRE*:
COUNT(H[name = "Hilton"].?.M[name = "Louvre" ∧ I.t_s ∧ $f_{Time}^{timeId \to TimeOfDay}(t_s)$ = "*morning*"])

In these queries, the conditions are evaluated over the *current* nodes (the node the parser is evaluating). For instance, in Q2, if the parser is at node 1 in Figure 3, the condition $name$ = "$Hilton$" returns "true" if $ext(1).name$ = "$Hilton$" and $label(1)$ = "$Museum$". Also, variable I corresponds to the time interval of the node that is being visited when evaluating the expression, and t_s is its starting point. The following query shows the full power of the language, because it includes geometric and temporal conditions, that show how all elements in the model interact. Here, also, the encoded fact table is not enough, and we need to access the geometry. However, note that for many useful queries and patterns, much simpler expressions will suffice. The query is: "*Total number of trajectories going from a tourist attraction to a museum in the 19th district of Paris in the morning.*"
COUNT(T.?.M[$f_{\text{Time}}^{\text{timeId}\to\text{TimeOfDay}}$(I.$t_s$) = "morning" ∧ (∃ g_{id}) (∃ x) (∃ y) (∃ O_{id}) ∃ (t_1) (∃ p) (∃ pg) (∃ d) ($M(O_{id}, t_1, x, y)$ ∧ $\alpha_{L_p,Museum}^{m_{id}\to Pg}(p) = g_{id}$ ∧ $r_{L_p}^{point\to Pg}(x,y) = g_{id}$ ∧ $\alpha_{L_d,Distr}^{distr\to Pg}(d) = p_g$ ∧ $pg.number = 19$ ∧ $f_{L_{dist}}^{point\to Pg}(x,y) = pg$])

Let us explain this expression. The function $\alpha_{L_p,Museum}^{m_{id}\to Pg}(p) = g_{id}$, maps the id of the PoI in the extension of the *current* node (p), to the polygon representing it in the geographic part (g_{id}), while $\alpha_{L_d,Distr}^{distr\to Pg}(d) = p_g$ has the meaning already explained. The equality $f_{L_{dist}}^{point,Pg}(x,y) = pg$ checks that the point of the trajectory belongs to the 19th district.

6. FUTURE WORK

Our future work will be focused in the implementation of the model and query languages proposed here [1]. We also believe that the *REsm* language is promising for mining trajectory data, specifically in the context of sequential patterns mining with constraints, and we will work in this direction.

7. ACKNOWLEDGMENTS

This work has been partially funded by the European Unionă under the FP6-IST-FET programme, Project n. FP6-14915, GeoPKDD: Geographic Privacy-Aware Knowledege Discovery and Delivery, the Research Foundation Flanders (FWO-Vlaanderen), Project G.0344.05., and the Scientific Agency of Argentina, Project PICT n. 21350.

[1]A preliminary implementation can be found in http://piet.exp.dc.uba.ar/moving

8. REFERENCES

[1] S. Brakatsoulas, D. Pfoser, and N. Tryfona. Modeling, storing and mining moving object databases. In *Proceedings of IDEAS'04*, pages 68–77, Washington D.C, USA, 2004.

[2] M. L. Damiani, J. A. Fernandes de Macedo, C. Parent, F. Porto, and S. Spaccapietra. A conceptual view of trajectories. *Technical Report, Ecole Polythecnique Federal de Lausanne, April 2007*, 2007.

[3] L. Gómez, S. Haesevoets, B. Kuijpers, and A. Vaisman. Spatial aggregation: Data model and implementation. In *Submitted for review*, 2006.

[4] R. H. Güting and M. Schneider. *Moving Objects Databases*. Morgan Kaufman, 2005.

[5] K. Hornsby and M. Egenhofer. Modeling moving objects over multiple granularities. *Special issue on Spatial and Temporal Granularity, Annals of Mathematics and Artificial Intelligence*, 2002.

[6] R. Kimball and M. Ross. *The Data Warehouse Toolkit: The Complete Guide to Dimensional Modeling, 2nd. Ed.* J.Wiley and Sons, Inc, 2002.

[7] B. Kuijpers and A. Vaisman. A data model for moving objects supporting aggregation. In *Proceedings of the First International Workshop on Spatio-Temporal Data Mining (STDM'07)*, Istambul, Turkey, 2007.

[8] Jae-Gil Lee, Jiawei Han, and Kyu-Young Whang. Trajectory clustering: a partition-and-group framework. In *SIGMOD Conference*, pages 593–604, Beijing, China, 2007.

[9] N. Meratnia and R. de By. Aggregation and comparison of trajectories. In *Proceedings of the 26th VLDB Conference*, Virginia, USA, 2002.

[10] C. Mouza and P. Rigaux. Mobility patterns. *Geoinformatica*, 9(23):297–319, 2005.

[11] Th. Ott and Fr. Swiaczny. *Time-integrative Geographic Information Systems–Management and Analysis of Spatio-Temporal Data*. Springer, 2001.

[12] D. Papadias, Y. Tao, J. Zhang, N. Mamoulis, Q. Shen, and J. Sun. Indexing and retrieval of historical aggregate information about moving objects. *IEEE Data Eng. Bull.*, 25(2):10–17, 2002.

[13] J. Paredaens, G. Kuper, and L. Libkin, editors. *Constraint databases*. Springer-Verlag, 2000.

[14] P. Rigaux, M. Scholl, and A. Voisard. *Spatial Databases*. Morgan Kaufmann, 2002.

[15] M. Vazirgiannis and O. Wolfson. A spatiotemporal model and language for moving objects on road networks. In *SSTD*, pages 20–35, Redondo Beach, CA, USA, 2001.

[16] O. Wolfson, P. Sistla, B. Xu, and S. Chamberlain. Domino: Databases fOr MovINg Objects tracking. In *Proceedings of SIGMOD'99*, pages 547 – 549, 1999.

[17] O. Wolfson, B. Xu, S. Chamberlain, and L. Jiang. Moving objects databases: Issues and solutions. In *SSDBM*, pages 111–122, Capri, Italy, 1998.