A Model for Enriching Trajectories with Semantic Geographical Information

Luis Otavio Alvares
Instituto de Informatica
UFRGS, Brazil
Hasselt University &
Transnational University of
Limburg, Belgium
alvares@inf.ufrgs.br

Vania Bogorny
Theoretical Computer Science
Hasselt University &
Transnational University of
Limburg, Belgium
vania.bogorny@uhasselt.be

Bart Kuijpers
Theoretical Computer Science
Hasselt University &
Transnational University of
Limburg, Belgium
bart.kuijpers@uhasselt.be

Jose Antonio Fernandes de Macedo
Ecole Polytechnique Federale
de Lausanne,
Switzerland
jose.macedo@epfl.ch

Bart Moelans
Theoretical Computer Science
Hasselt University &
Transnational University of
Limburg, Belgium
bart.moelans@uhasselt.be

Alejandro Vaisman
Universidad de Buenos Aires
Argentina
avaisman@dc.uba.ar

ABSTRACT

The collection of moving object data is becoming more and more common, and therefore there is an increasing need for the efficient analysis and knowledge extraction of these data in different application domains. Trajectory data are normally available as sample points, and do not carry semantic information, which is of fundamental importance for the comprehension of these data. Therefore, the analysis of trajectory data becomes expensive from a computational point of view and complex from a user's perspective. Enriching trajectories with semantic geographical information may simplify queries, analysis, and mining of moving object data. In this paper we propose a data preprocessing model to add semantic information to trajectories in order to facilitate trajectory data analysis in different application domains. The model is generic enough to represent the important parts of trajectories that are relevant to the application, not being restricted to one specific application. We present an algorithm to compute the important parts and show that the query complexity for the semantic analysis of trajectories will be significantly reduced with the proposed model.

Categories and Subject Descriptors

H.2.8 [**Database Applications**]: Spatial Databases and GIS

General Terms

Design, Algorithms, Theory

Keywords

Semantic Trajectories, Stops and Moves, Trajectory Data Analysis, Moving Objects, Querying Trajectories

1. INTRODUCTION

The collection of moving object data has become common in the recent years, and therefore there is an increasing necessity to provide mechanisms for the efficient analysis and knowledge extraction from these data.

Moving object data are normally available as sample points in the form (tid, x, y, t), where tid is an object identifier and x, y and t are respectively spatial coordinates and a time stamp. The integration of trajectory data with semantic geographical information is the main step for trajectory data analysis in real applications.

Several data models have been proposed for efficiently querying trajectory sample points, such as [6, 8, 10, 19]. A few prototypes of spatio-temporal database management systems have been developed [5, 14] to provide the operations to manipulate spatio-temporal data, but the integration of trajectories with the relevant geographic information is still user dependent, and has to be performed on the fly for each query, as shown in Figure 1.

Trajectories and geographic data overlap in space, and therefore their integration is the first step toward trajectory data analysis. According to Brakatsoulas et al. [3], the analysis of trajectory data consists of the integration of spatial, non-spatial, and trajectory data. This integration is application dependent, where the user specifies the spatial feature types (e.g., hotels, touristic places) that are relevant to the analysis of trajectories. It is well known that the spatial join is the bottleneck in spatial data analysis, but it is the basis to answer any spatio-temporal query, as we will explain in more detail in Section 1.1.

Another problem is a lack of semantic analysis for which more complex queries are necessary. To answer such queries

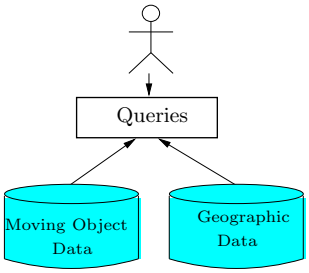

Figure 1: Current framework for trajectory data analysis

normally more sophisticated methods like data mining algorithms may be necessary. For example, (i) *which are the places most frequently visited by people attending a conference in a touristic city*?; (ii) *which are the main sequences of places visited in the morning*?; (iii) *which are the places that moving objects pass through and stay for a certain amount of time*? Several queries about moving behavior can only be answered by considering trajectories as well as their semantics [1], as we will explain in more detail in Section 1.2.

1.1 The Problem of Query Formulation and Time Complexity

We introduce the problem of spatial joins in trajectory data analysis by a simple query example.

Q1: Which are the places that moving object A has passed during its trajectory, assuming that each object has only one trajectory in the database and that the interesting geographic places are *hotels* and *touristic places*?

Let us consider that the moving object A has the trajectory (1), shown in Figure 2, which has no semantic geographic information, and is represented as a set (x,y,t). In this scenario, Q1 will be similar to the query bellow, where two spatial joins are necessary to answer this query:

```
SELECT h.name
FROM trajectory t, hotel h
WHERE t.id='A' AND
     intersects(t.movingpoint.geometry,h.geometry)
UNION
SELECT p.name
FROM trajectory t, touristicPlace p
WHERE t.id='A' AND
     intersects(t.movingpoint.geomtetry,p.geometry)
```

This query is quite simple, and considering a trajectory as a set of sample points, the intersection operation would be tested until the first point in the trajectory intersects the given places (hotel and/or touristic place). The complexity increases when the query includes time. For instance, *which are the places that an object* A *has stayed for more than three hours*? In spatio-temporal queries the complexity increases in both formulation and computational time, because the search cannot stop when the first point of the trajectory intersects a given place. All points in the trajectory that

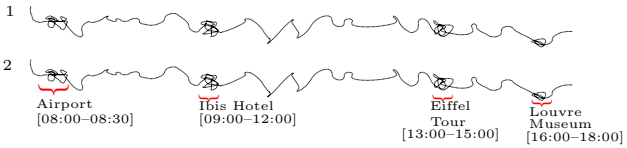

Figure 2: (1) Single raw trajectory and (2) single semantic trajectory

intersect the place have to be tested until the moving object leaves the place, or until the time constraint is satisfied.

Now let us consider that object A has the *semantic trajectory* (2), shown in Figure 2, where the integration of the geographic information with the trajectory was performed in a preprocessing step, and the user has defined all geographic places that are relevant to the application. In this scenario, Q1 will be something like:

```
SELECT t.place
FROM semanticTrajectory t
WHERE  t.id='A'
```

On the one hand, the *semantic trajectory* shown in Figure 2 (2), aggregates the semantic geographic information that is necessary for the analysis of the trajectory, and this information can be *reused* in any query. On the other hand, in the trajectory without semantics shown in Figure 2(1), the relationships of the trajectory with the relevant geographic objects have to be *recomputed* in every different query, either for the same or different geographic object types.

Besides the problem of spatial joins, there is a lack of semantic analysis at both representation and data manipulation levels that requires an a priori integration of these data [12], as will be explained in the following section.

1.2 The Problem of Trajectory Data Analysis

The a priori integration of trajectory data with the background geographic information may lead to the discovery of semantic trajectory patterns that many data mining techniques [4, 9, 11, 14, 16] that consider trajectories as sample points (tid, x, y, t) may not be able to discover. An example is shown in Figure 3. On the left side a set of trajectories is represented in the form of sample points, without semantics. On the right side, the geographic information is integrated to trajectories. While over the sample points no patterns can be visualized, over the semantic trajectories shown in Figure 3 (right) three semantic patterns can be inferred among the four trajectories. (1) Three trajectories have a move from Hotel (H) to Touristic Place (TP); (2) The three trajectories return from Touristic Place to the Hotel from where they go to the Conference Center (CC); and (3) all four trajectories move from Hotel to the same Conference Center.

Notice in Figure 3 (right) that semantic patterns or semantic relationships are independent of (x,y) coordinates. These patterns are sparse in space, and would not be identified by considering only the geometric properties of the trajectories. The hotels and touristic places, for instance, are *not* lo-

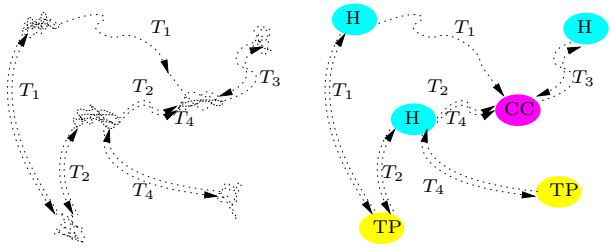

Figure 3: (left) Set of trajectory sample points and (right) set of semantic trajectories

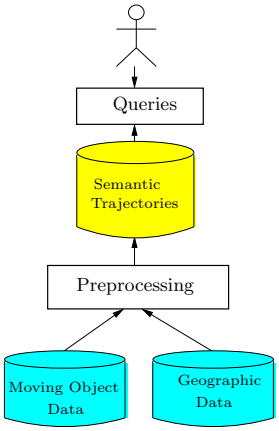

Figure 4: Framework for semantic trajectories

cated in *dense regions*, which is the measure adopted in most spatio-temporal data mining algorithms to compute trajectory patterns. This example shows that semantics plays an essential role in trajectory data analysis and knowledge extraction.

1.3 Scope and Outline

In this paper we propose a trajectory data preprocessing method to integrate, off-line, trajectories with the geographic information that is relevant to the application. For this purpose, we adopt the trajectory data model proposed by [15], in which the user has the vision on trajectories as a set of *stops* and *moves*. Stops are the important places of the trajectory where the object has stayed for a minimal amount of time. Following this model we will generate a semantic trajectory dataset, extracted from trajectory sample points, as shown in the framework in Figure 4. As a consequence, the user will perform the queries over semantic trajectories.

The scope of this paper is limited to the formal definition of semantic trajectories, the presentation of an algorithm to create a semantic trajectory dataset, and an empirical analysis to compare semantic trajectories and trajectories represented as sample points. The remainder of the paper is organized as follows: In Section 2, we present the formal model to represent trajectories with semantic geographic information as well an algorithm to implement the model. In Section 3, we present an empirical analysis to show the application and usability of our approach. In Section 4, the related works and our contributions are presented. Section 5 concludes the paper and present directions of future work.

2. THE SEMANTIC DATA MODEL

In this section we present a formal model to represent semantic trajectories, using stops and moves to integrate geographic information to trajectory sample points.

2.1 Trajectories and Trajectory Samples

Let $\mathbf{R}$ denote the set of real numbers. We restrict ourselves to movement in the real plane $\mathbf{R}^2$. Space-time space will be denoted $\mathbf{R}^2 \times \mathbf{R}$, where the first two dimensions represent space and the latter represents time. Typically, we will use x, y as variables that range over spatial coordinates and t as a variable that ranges over time points.

DEFINITION 1. *A* sample trajectory *is a list of space-time points* $\langle (x_0, y_0, t_0), (x_1, y_1, t_1), \ldots, (x_N, y_N, t_N) \rangle$, *where* $x_i, y_i, t_i \in \mathbf{R}$ *for* $i = 0, \ldots, N$ *and* $t_0 < t_1 < \cdots < t_N$. □

For the sake of finite representability, we may assume that the space-time points (x_i, y_i, t_i), have rational coordinates. This will be the case in practice, since these points are typically the result of observations.

2.2 Stops and Moves

In the remainder of this work, if we talk about (raw) trajectories, we assume they are given as a sample trajectory as described in Definition 1.

In this section, we define what the stops and moves of a trajectory are. This definition is dependent on the particular application one is interested in. First, we define the notions of candidate stops and application.

2.2.1 Candidate stops

DEFINITION 2. *A* candidate stop C *is a tuple* (R_C, Δ_C), *where* R_C *is a (topologically closed) polygon in* $\mathbf{R}^2$ *and* Δ_C *is a strictly positive real number. The set* R_C *is called the* geometry *of the candidate stop and* Δ_C *is called its* minimum time duration.

An application $\mathcal{A}$ *is a finite set* $\{C_1 = (R_{C_1}, \Delta_{C_1}), \ldots, C_N = (R_{C_N}, \Delta_{C_N})\}$ *of candidate stops with mutually non-overlapping geometries* $R_{C_1}, \ldots, R_{C_N}$ □

In case that a candidate stop is a point or a polyline, we will generate a polygonal buffer around this object, and thus represent it as a polygon in the application, in order to overcome spatial uncertainty.

2.2.2 Stops and Moves of a Trajectory

DEFINITION 3. *Let* T *be a trajectory and let*

$$\mathcal{A} = (\{C_1 = (R_{C_1}, \Delta_{C_1}), \ldots, C_N = (R_{C_N}, \Delta_{C_N})\})$$

be an application.

Suppose we have a subtrajectory $\langle (x_i, y_i, t_i), (x_{i+1}, y_{i+1}, t_{i+1}), \ldots, (x_{i+\ell}, y_{i+\ell}, t_{i+\ell}) \rangle$ *of* T, *where there is a* (R_{C_k}, Δ_{C_k}) *in* $\mathcal{A}$ *such that* $\forall j \in [i, i+\ell] : (x_j, y_j) \in R_{C_k}$ *and* $|t_{i+\ell} - t_i| \geq \Delta_{C_k}$, *and this subtrajectory is maximal (with respect to these two conditions), then we define the tuple* $(R_{C_k}, t_i, t_{i+\ell})$ *as a* stop of T with respect to $\mathcal{A}$.

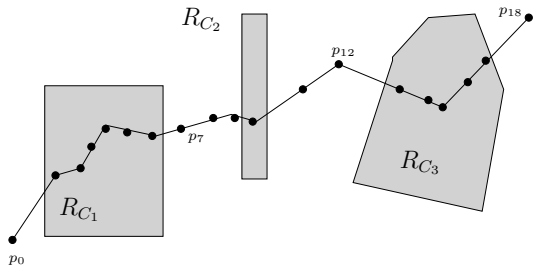

Figure 5: Example of application with three candidate stops

A move of T with respect to $\mathcal{A}$ *is one of the following cases:*

- *a maximal contiguous subtrajectory of T in between two temporally consecutive stops of T;*
- *a maximal contiguous subtrajectory of T in between the initial point of T and the first stop of T;*
- *a maximal contiguous subtrajectory of T in between the last stop of T and the last point of T;*
- *the trajectory T itself, if T has no stops.*

When a move starts in a stop, it starts in the last point of the subtrajectory that intersects the stop. Analogously, if a move ends in a stop, it ends in the first point of the subtrajectory that intersects the stop. □

Figure 5 illustrates these concepts. In this example, there are three candidate stops with geometries R_{C_1}, R_{C_2}, and R_{C_3}. Let us imagine that the spatial projection of the trajectory T is run through from left to right and $t_0, \ldots, t_{18}$ are the time points of T. First, T is outside any candidate stop, so we start with a move. Then T enters R_{C_1} at time point t_1. Since the duration of staying inside R_{C_1} is long enough, (R_{C_1}, t_1, t_6) is the first stop of T. Next, T enters R_{C_2}, but for a time interval shorter than Δ_{C_2}, so this is not a stop. We therefore have a move until T enters R_{C_3}, which fulfills the requests to be a stop, and so $(R_{C_3}, t_{13}, t_{17})$ is the second stop of T. The trajectory T ends with a move.

In our definitions, stops are interesting spatial locations, also called spatial features, specified according to the application. For instance, traffic lights may be considered as stops in a transportation management application, but probably not in a tourism application. Spatial features are normally stored in different files (e.g. shape files) or in different relations (e.g., hotel, airport) in geographic databases. Therefore, it is possible to join trajectory sample points with important spatial features, in order to find stops and moves, as will be explained in the following section.

2.3 An Algorithm to Extract Stops and Moves

According to the formalisms defined in the previous section, we describe in Listing 1, in pseudo code, an algorithm to find stops and moves, which we call SMoT (Stops and Moves of Trajectories).

In general words, the algorithm verifies for each point of a trajectory T if it intersects the geometry of a candidate stop R_C (line 12 and line 13). In affirmative case, the algorithm looks if the duration of the intersection is at least equal to a given threshold Δ_C (line 22). If this is the case, the intersected candidate stop is considered as a stop, and this stop is recorded. Note that when we use T or R_C as parameter (e.g., in line 23), we use and assign in reality an identifier (foreign key), and not the whole data structure. We do not need to store the geometry of the stop because we store the instance and the type of the spatial feature in which the stop occurs.

Listing 1: SMoT pseudo-code

```
1  INPUT: T //set of trajectories
2         A //application
3
4  OUTPUT: S //set of stops
5          M //set of moves
6
7  METHOD:
8  S=new Stops(); M=new Moves();
9  FOR each trajectory T ∈ T DO
10  i=0; previousStop = null;
11  WHILE(i ≤ size(T)) DO
12    IF (∃(R_C, Δ_C) ∈ A |
13       geometry(T[i]) intersects R_C)
14        //using spatial index
15    enterTime = time(T[i]);  i++;
16    WHILE (intersects(T[i],R_C)) DO
17     i++;
18    ENDWHILE
19    //Go one step back (went outside R_C)
20    i--;
21    leaveTime = time(T[i]);
22    IF (leavetime-entertime >= Δ_C)
23     stop = (T,R_C,enterTime,leaveTime);
24     S.add(stop);
25     move=(T,previousStop,stop,
26          previousStop.leaveTime,enterTime)
27     M.add(move);
28     previousStop = stop;
29    ENDIF
30   ENDIF
31   i++;
32   j=1;
33   WHILE((i+j ≤ size(T))
34          and (T[i+j]-T[i]<min_{Δ_C}(A))) DO
35        j++;
36   ENDWHILE
37   IF (∄((R_C, Δ_C) ∈ A |
38         geometry(T[i+j-1]) intersects R_C)
39    i=i+j;
40   ENDIF
41  ENDWHILE
42  IF (T[i-1] not ∈ previousStop)
43   //T do not end with a stop
44   move=(T,previousStop,null,
45        previousStop.leaveTime,time(T[i-1]))
46   M.add(move);
47  ENDIF
48 ENDFOR
```

Table 1(b) shows an example of a stop dataset where the attribute *Tid* corresponds to the trajectory identifier, *Sid* is the stop identifier, *SFTid* corresponds to the instance of the spatial feature type, *SFTname* represents the type of the spatial feature where the stop occurs (e.g. airport, hotel), and *Sbegint* and *Sendt* represent the duration time of

the stop. The pair *SFTid* and *SFTname* corresponds to the name of the stop, and is the pointer to the relevant feature in which the stop occurs. Notice that having access to the spatial feature type in which the stop occurs allows the user to perform several semantic queries on the non-spatial attributes over the spatial feature that corresponds to a stop.

A move is recorded between the previous stop and the latest one. This previous stop can be null, if the latest stop is the first stop of the trajectory. When a move m is inserted into the set of moves M (line 27), its space-time characteristics are also added. While the stops are inside a spatial feature type that has a geometry, the intersection of the geometry of a move is not tested with any spatial feature type because moves are not considered as important parts of the trajectory. However, for some applications, it might be interesting to know the spatial features that a move has crossed, and therefore we keep the geometry and the time stamp of the move for further spatial analysis, which are out of the scope of this work.

Table 1(c) shows an example of a move dataset, where *Tid* is the trajectory identifier, *Mid* is the move identifier, *S1id* and *S2id* are respectively the two stops in which the move occurs, and *geometry* and *timest* correspond to the moving point of the move.

In our algorithm, we exploit the functionalities of the spatial database by using the available spatial index to check if there exists a candidate stop that intersects with a vertex of the trajectory (line 12 and line 13). We also take advantage (line 32–40) of the fact that the minimum stop duration has to be Δ_C (see definition 2). If we are in point $p_i \in T$ with time point t_i, and p_i is located on a move or at the beginning of a candidate stop, we can take a look at point $p_j \in T$ with time point t_j such that $t_j - t_i < min_{\Delta_C}(\mathcal{A})$. If that p_j is not located inside a candidate stop we do not have to check the points between p_i and p_j on T because they can never be part of a stop.

In summary, the output of SMoT is a semantic trajectory dataset, and therefore different semantic trajectory analysis may be performed. In the next section we present some empirical analysis over semantic trajectories and trajectory sample points.

3. ANALYSIS

The a priori integration of trajectories with semantic geographic information that characterizes the most important parts (places) of a trajectory according to the application does significantly reduce the complexity of the query and facilitates trajectory data analysis.

The stops and moves are computed only once, in a preprocessing step, and therefore, the spatial search space and spatial joins in the query formulation are minimized, in relation to the model of sample points. The decomposition of trajectories into stops and moves provides direct access to the semantic trajectory information. As can be observed in the example shown in Table 1(b), stops become a relational model with a foreign key to the instance of the relevant spatial feature type, represented in Table 1(d) and Table 1(e). Similarly, the moves in Table 1(c) have a foreign key to the stop where each move starts (S1id) and finishes(S2id).

Table 1: Example Datasets

(a) trajectory sample point

Tid	geometry	timest
1	48.890018 2.246100	08:25
1	48.890018 2.246100	08:26
...	...	...
1	48.890020 2.246102	08:40
1	48.888880 2.248208	08:41
1	48.885732 2.255031	08:42
...	...	...
1	48.858434 2.336105	09:04
1	48.853611 2.349190	09:05
...	...	...
1	48.853610 2.349205	09:40
1	48.860515 2.349018	09:41
...	...	...
1	48.861112 2.334167	10:00
1	48.861531 2.336018	10:01
1	48.861530 2.336020	10:02
...	...	...
2	...	...

(b) Stop

Tid	Sid	SFTid	SFTname	Sbegint	Sendt
1	1	1	Hotel	08:25	08:40
1	2	1	TouristicPlace	09:05	09:30
1	3	3	TouristicPlace	10:01	14:20
...	...	...	...	...	...

(c) Move

Tid	Mid	S1id	S2id	geometry	timest
1	1	1	2	48.888880 2.246102	08:41
1	1	1	2	48.885732 2.255031	08:42
...	...	...	...	...	...
1	1	1	2	48.860021 2.336105	09:04
1	2	2	3	48.860515 2.349018	09:41
...	...	...	...	...	...
1	2	2	3	48.861112 2.334167	10:00
...	...	...	...	...	...

(d) Hotel

Id	Name	Stars	geometry
1	Ibis Nanterre	2	48.890015 2.246100, ...
2	Meridien	5	48.880005 2.283889, ...
...	...	...	...

(e) Touristic Place

Id	Name	Type	geometry
1	Notre Dame	Church	48.853611 2.349167, 48.853612 2.350556,..
2	Eiffel Tower	Monument	48.858330 2.294333, 48.858055 2.289444,..
3	Louvre	Museum	48.862220 2.335556, 48.860833 2.339722,..
.	...	...	...

In our examples shown in Table 1 we have considered a spatial database, in which the algorithm is being implemented and evaluated. Therefore the moving points of trajectory samples (Table 1(a)) and moves (Table 1(c)) are represented in two different attributes (geometry and timest).

In this section we will analyze some query examples using the semantic trajectory model presented in the previous section, and make a comparison with the model of sample points, in a tourism application. For this analysis we will consider the tables shown in Table 1, and that the relevant spatial feature types for the application are Hotels and Touristic Places, shown in Table 1(d) and Table 1(e).

Q2: How many trajectories go from a hotel to at least one touristic place?

In this question there is a sequence of movements that has to be taken into account, where hotel is before a touristic place, so the time has to be considered in the query. Considering

trajectories as sample points, a query similar to the following would be performed.

```
SELECT distinct count(t.Tid)
FROM trajectory t, trajectory u,
     hotel h, touristicPlace p
WHERE intersects (t.geometry, h.geometry) AND
      intersects (u.geometry, p.geometry) AND
      t.Tid=u.Tid AND u.timest>t.timest
```

In this model, two spatial joins are necessary to test the intersection of the trajectories with both hotels and touristic places. The time has to be tested in order to validate the sequence of the movement.

Following the model of stops and moves to answer this query neither spatial join nor time verification is necessary. The sequence (order) of stops is represented by the Stop identifier, and the test $a.Sid < b.Sid$ in the query using our model will give the order of the stops in time.

```
SELECT distinct count(a.Tid)
FROM stop a, stop b
WHERE a.SFTname='Hotel' AND
      b.SFTname='Touristic Place' AND a.Tid=b.Tid
      AND a.Sid < b.Sid
```

Q3: How many trajectories visit the Notre Dame church and then visit the Pompidou Center crossing the Arcole bridge?

Using the trajectory sample points, to answer this question at least three spatial joins are necessary, and the order as the intersections occur has to be Notre Dame, Arcole bridge, and Pompidou Center, in this order. All trajectories have to be spatially tested with the geographic object types specified in the query, as well as the time constraint to check the order of the move. The query will be similar to the following.

```
SELECT distinct count(t.Tid)
FROM trajectory t, trajectory u, trajectory v,
     touristicPlace p, touristicPlace q, bridge b
WHERE intersects (t.geometry, p.geometry) AND
      p.Name='Notre Dame' AND
      intersects (u.geometry, q.geometry) AND
      q.Name='Pompidou Center' AND
      intersects(v.geometry, b.geometry) AND
      b.Name='Arcole' AND
      t.Tid=u.Tid AND t.Tid=v.Tid AND
      t.timest < v.timest AND v.timest < u.timest
```

Using the model of stops and moves, only one spatial join operation is necessary, and for one specific move (from stop Notre Dame to stop Pompidou Center). It is the geometry of the move that will be used to test the intersection with Arcole bridge, which in our application is not a stop because bridges were not defined as an interesting place. The search space on which the join will be applied is significantly reduced, since the moves are filtered by the stop constraint, i.e., only the moves between the two specific stops Notre Dame and Pompidou Center will be tested, as shows the following query.

An important remark is that when testing the moves, the order of the move is given by the attributes *S1id* and *S2id*. This avoids the necessity to check the time constraint inside a move. This query example should not be usual, i.e., where the geometry of the move will be used, since the most important parts of a trajectory are the stops. However, it illustrates that any further analysis over moves is still possible because moves contain the spatio-temporal characteristics of the trajectory.

```
SELECT distinct count(a.Tid)
FROM move m, touristicPlace p,
     touristicPlace q, bridge b
WHERE m.S1id=p.Id AND p.Name='Notre Dame' AND
      m.S2id=q.Id AND q.Name='Pompidou Center' AND
      intersects(m.geometry, b.geometry) AND
      b.Name='Arcole'
```

Q4: Which are the touristic places that moving objects have passed and stayed for more than one hour?

To answer question Q4 considering the sample points approach, a more sophisticated query is necessary, but to simplify it we assume that: (i) the interval between samples is about one minute and (ii) if one touristic place has been visited more than once, we count the total amount of time. This query will be similar to the following:

```
SELECT temp.name, count(*) AS n_visits
FROM (SELECT t.Tid, p.name
      FROM trajectory t, touristicplace p
      WHERE intersect(t.geometry,p.geometry)
      GROUP BY t.Tid,p.name
      HAVING count(*)>60) AS temp
GROUP BY temp.name
```

Considering our approach, a simplified query can be formulated, such as:

```
SELECT s.SFTname, count(*) AS n_visits
FROM stop s, touristicplace p
WHERE s.SFTid=p.id AND (s.Sendt - s.Sbegint ) > 60
GROUP BY s.SFTname
```

A question like Q4 is still quite simple to be answered using the sample points approach, since it refers to only one type of important place (touristic places). However, a very similar query like *which are all important places that the moving objects have passed and have stayed for more than one hour* would become more complex, and similar to our algorithm to extract stops and moves.

As we have seen in only a few examples, the proposed model of stops and moves may require some time to be computed, but much less time will be required by the user for the analysis over trajectories.

Besides reducing the search space of the trajectory queries, the transformation of trajectory sample points into stops and moves allows the semantic exploration of the non-spatial attributes of all spatial features that represent a stop. For instance, select all trajectories that have a stop at a two stars hotel, or select all trajectories that stop at a two stars hotel and also stop at touristic museums.

More sophisticated analysis over trajectories may require data mining methods. Therefore, our model of stops and moves becomes even more powerful, and may considerably simplify the use of data mining algorithms. Since trajectory data mining is out of the scope of this paper, we will illustrate the usability of our model for data mining with a very simple example.

Q5: In relation to moving behavior, which is the most frequent sequence of two important places followed by the moving objects?

While for existing trajectory data mining algorithms that deal with sample points like [4, 14, 11, 16], such a question may be complex to answer, using the model of stops and moves a query like the following would answer this question:

```
SELECT S1id, S2id
FROM move
GROUP BY S1id, S2id
HAVING MAX(COUNT(S1id,S2id)) =
          ( SELECT MAX(COUNT(S1id, S2id))
            FROM move
            GROUP BY S1id, S2id)
```

The model of stops and moves allows the use of traditional data mining methods like association rules and frequent pattern mining.

In this section we evaluated the proposed model with a few query examples, focusing on a tourism application. However, as we have pointed out at the beginning of the paper, the geographic information to be integrated to trajectories is application dependent, and therefore, our semantic model of stops and moves is general enough to support different applications. In applications like traffic management, for instance, candidate stops could be traffic lights, bus stops, train stations, round abouts, etc. In a urban planning application, for instance, for cultural and recreational domain, candidate stops could be parks, lakes, recreational areas, shopping centers, parking places, bus stops, etc.

4. RELATED WORKS

Moving object data have received significant attention in the last few years from the database community. Güting [6, 8] has proposed several data types and operations to manipulate moving object data, which have been implemented in a moving object database prototype [5]. Similarly, [19] has proposed operations to manipulate moving objects. Based on these definitions, [14] has developed the HERMES prototype, which is a new data cartridge that exploits the spatial data types available in Oracle.

Main research approaches on moving object databases have focused on the geometrical and temporal characteristics of trajectory sample points, but little attention has been devoted for a more semantic representation of trajectories from an application point of view [15]. The necessity for the integration of geographic information and trajectories has been expressed in a few works that focus on the network application domain [3, 7, 12, 17]. According to Güting [7], it makes more sense to describe movements relative to the network rather than unconstrained space, because then it is much easier to formulate queries between moving objects and the network.

The conceptual model proposed by [15] integrates geographic information to trajectories through the stops. Similarly, in [3] and [12] the *key points* of trajectories are extracted, but specifically for an application where the background geographic information refers to the road network. Rigaux [13] has proposed a model where trajectories are represented as a sequence of moves between regions.

As we have seen in the analysis section, the decomposition of trajectory sample points into stops and moves reduces the computational complexity of most queries from a spatial query to a conjunctive query. From the user's perspective, the semantic model may facilitate the formulation of queries and facilitate trajectory data analysis. In summary, our approach differs from existing ones in the following aspects:

- We provide a general model, which is application independent, to integrate trajectory data and geographic information in a preprocessing step. In other words, we propose a model to automatically add semantic information to trajectories based on the notion of stops and moves introduced in [15].
- We present the algorithm SMoT to extract stops and moves.
- Through an empirical analysis we show that the enrichment of trajectories with semantic information in a preprocessing step facilitates the query formulation and more powerful analysis can be performed over trajectories.

5. CONCLUSIONS AND FUTURE WORKS

Trajectory data are normally available in the form of sample points, what makes their analysis in different application domains expensive from a computational point of view and complex from a user's perspective. For a very simple query like *which is the set of given important places that objects have passed during a given time interval* may require a complex query, with several spatial joins among trajectories and geographic feature types. In a semantic trajectory dataset a single query over the table Stops would answer such query.

The proposed method for the analysis of trajectory data can take some time to add the semantic information (the extraction of stops and moves), but much less time will be required from the user for querying and analyzing trajectories. By computing stops and moves in one preprocessing step, the complexity of most queries is reduced from a spatial query to a conjunctive query.

The transformation of trajectory sample points into stops and moves reduces the search space in two main parts: (i) stops, where a set of sample points is transformed into a geographic object that has a meaning; (ii) moves, where the sample points are reduced to a small part of a trajectory between two stops.

Future ongoing work is the implementation of the model of stops in moves into Weka [18], which is a free and open source data mining toolkit that we have extended to support automatic geographic data preprocessing for spatial data mining [2].

Acknowledgments

This research has been funded by the Brazilian agency CAPES (BEX 3631/06-0), the European Union (FP6-IST-FET programme, Project n. FP6-14915, GeoPKDD: Geographic Privacy-Aware Knowledge Discovery and Delivery), the Research Foundation Flanders (FWO-Vlaanderen), Research Project G.0344.05, and Scientific Agency of Argentina (Project PICT n. 21350).

6. REFERENCES

[1] Luis Otavio Alvares, Vania Bogorny, Jose Fernandes de Macedo, and Bart Moelans. Dynamic modeling of trajectory patterns using data mining and reverse engineering. In *26th International Conference on Conceptual Modeling - ER2007 - Tutorials, Posters, Panels and Industrial Contributions*, volume 83. CRPIT (to appear), November 2007.

[2] Vania Bogorny, Andrey Palma Tietbhol, Paulo Engel, and Luis Otavio Alvares. Weka-gdpm: Integrating classical data mining toolkit to geographic information systems. In *WAAMD*, pages 9–16. SBC, 2006.

[3] Sotiris Brakatsoulas, Dieter Pfoser, and Nectaria Tryfona. Modeling, storing, and mining moving object databases. In *IDEAS '04: Proceedings of the International Database Engineering and Applications Symposium (IDEAS'04)*, pages 68–77, Washington, DC, USA, 2004. IEEE Computer Society.

[4] Joachim Gudmundsson and Marc van Kreveld. Computing longest duration flocks in trajectory data. In *GIS'06: Proceedings of the 14th annual ACM international symposium on Advances in geographic information systems*, pages 35–42, New York, NY, USA, 2006. ACM Press.

[5] Ralf Hartmut Güting, Victor Almeida, Dirk Ansorge, Thomas Behr, Zhiming Ding, Thomas Hose, Frank Hoffmann, Markus Spiekermann, and Ulrich Telle. Secondo: An extensible dbms platform for research prototyping and teaching. In *ICDE '05: Proceedings of the 21st International Conference on Data Engineering (ICDE'05)*, pages 1115–1116, Washington, DC, USA, 2005. IEEE Computer Society.

[6] Ralf Hartmut Güting, Michael H. Bhlen, Martin Erwig, Christian S. Jensen, Nikos A. Lorentzos, Markus Schneider, and Michalis Vazirgiannis. A foundation for representing and querying moving objects. *ACM Trans. Database Syst.*, 25(1):1–42, 2000.

[7] Ralf Hartmut Güting, Victor Teixeira de Almeida, and Zhiming Ding. Modeling and querying moving objects in networks. *VLDB J.*, 15(2):165–190, 2006.

[8] Ralf Hartmut Güting and Markus Schneider. *Moving Objects Databases (The Morgan Kaufmann Series in Data Management Systems)*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2005.

[9] Panos Kalnis, Nikos Mamoulis, and Spiridon Bakiras. On discovering moving clusters in spatio-temporal data. In Claudia Bauzer Medeiros, Max J. Egenhofer, and Elisa Bertino, editors, *SSTD*, volume 3633 of *Lecture Notes in Computer Science*, pages 364–381. Springer, 2005.

[10] Bart Kuijpers and Walied Othman. Trajectory databases: Data models, uncertainty and complete query languages. In Thomas Schwentick and Dan Suciu, editors, *ICDT*, volume 4353 of *Lecture Notes in Computer Science*, pages 224–238. Springer, 2007.

[11] Patrick Laube, Stephan Imfeld, and Robert Weibel. Discovering relative motion patterns in groups of moving point objects. *International Journal of Geographical Information Science*, 19(6):639–668, 2005.

[12] Xiang Li, Christophe Claramunt, Cyril Ray, and Hui Lin. A semantic-based approach to the representation of network-constrained trajectory data. In *Progress in Spatial Data Handling - 12th International Symposium on Spatial Data Handling*, pages 451–464, Berlin, 2006. Springer.

[13] Cedric Mouza and Philippe Rigaux. Mobility patterns. *GeoInformatica*, 9:297–319(23), December 2005.

[14] Nikos Pelekis, Yannis Theodoridis, Spyros Vosinakis, and Themis Panayiotopoulos. Hermes - a framework for location-based data management. In Yannis E. Ioannidis, Marc H. Scholl, Joachim W. Schmidt, Florian Matthes, Michael Hatzopoulos, Klemens Böhm, Alfons Kemper, Torsten Grust, and Christian Böhm, editors, *EDBT*, volume 3896 of *Lecture Notes in Computer Science*, pages 1130–1134. Springer, 2006.

[15] Stefano Spaccapietra, Christine Parent, Maria-Luisa Damiani, Jose Antonio Fernandes de Macedo, Fabio Porto, and Christelle Vangenot. A conceptual view on trajectories. Technical report, Ecole Polytechnique Federal de Lausanne, April 2007.

[16] Ilias Tsoukatos and Dimitrios Gunopulos. Efficient mining of spatiotemporal patterns. In Christian S. Jensen, Markus Schneider, Bernhard Seeger, and Vassilis J. Tsotras, editors, *SSTD*, volume 2121 of *Lecture Notes in Computer Science*, pages 425–442. Springer, 2001.

[17] Michalis Vazirgiannis and Ouri Wolfson. A spatiotemporal model and language for moving objects on road networks. In Christian S. Jensen, Markus Schneider, Bernhard Seeger, and Vassilis J. Tsotras, editors, *SSTD*, volume 2121 of *Lecture Notes in Computer Science*, pages 20–35. Springer, 2001.

[18] Ian Witten and Eibe Frank. *Data Mining: Practical Machine Learning Tools and Techniques*. Morgan Kaufmann Series in Data Management Systems. Morgan Kaufmann, second edition, June 2005.

[19] Ouri Wolfson, Bo Xu, Sam Chamberlain, and Liqin Jiang. Moving objects databases: Issues and solutions. In Maurizio Rafanelli and Matthias Jarke, editors, *SSDBM*, pages 111–122. IEEE Computer Society, 1998.