Hybrid queries over symbolic and spatial trajectories: a usage scenario

Maria Luisa Damiani & Hamza Issa
University of Milan
Milan, Italy
E-mail:{maria.damiani,hamza.issa}@unimi.it

Ralf Hartmut Güting & Fabio Valdés
FernUniversität in Hagen
Hagen, Germany
E-mail:{fabio.valdes,rhg}@fernuni-hagen.de

***Abstract*—Symbolic trajectories is a novel data model recently proposed for the modeling and querying of temporally annotated sequences of symbolic descriptions, representing e.g. transportation means, places of interest, and so forth. Unlike geometric trajectories, symbolic trajectories capture the thematic dimension of movement. In this demonstration, we illustrate a practical approach to the querying of hybrid trajectories, combining the symbolic and geometric dimension in a multi-dimensional trajectory. The system runs on the Secondo moving object database. The multi-dimensional trajectories are obtained from the GeoLife dataset.**

***Keywords*—Semantic trajectories, mobility data, patterns**

I. Introduction

Mobility data modeling is gaining new popularity due to the growing concern for the semantic and contextual dimension of movement [1]. In line with this trend, the *symbolic trajectories* data model has been recently proposed to represent temporally annotated sequences of symbolic descriptions (labels) [2], [3]. Symbolic trajectories can be used, for example, to represent the activities performed by an individual in time, the transportation means used for moving from home to work, the attractions visited during a touristic trip. Formally based on the moving object data model [4], a symbolic trajectory is represented by an object of type *moving label*, denoting a time-varying string. Symbolic trajectories can be interrogated using a pattern-based query language enabling the extraction of sub-trajectories and possibly their classification.

Yet, symbolic trajectories are orthogonal to spatial trajectories. While this ensures a broad application potential, even outside the mobile domain, the existence of diverse and temporally interrelated trajectories, describing the movement from different angles - spatial, symbolic, numeric - complicates the analysis of mobility data. This motivates the concern for the notion of *multi-dimensional trajectories*. A multi-dimensional trajectory is a set of temporally correlated trajectories, possibly over different domains, called *dimensions*.

In this paper we present a practical approach to the handling of multi-dimensional trajectories. We focus in particular on the querying of trajectories defined over time intervals and consisting of one symbolic dimension and one spatial dimension. For the sake of generality, the dimensions are not temporally aligned (for example can be acquired from different sources or derived from other dimensions). The goal is to provide a flexible mechanism for composing and querying intertwined dimensions without affecting the underlying data models. This mechanism has been developed on top of Secondo, an extensible database system supporting the specification of new operators and data types, providing as well a large repertoire of algebras, including the moving object algebra extended with symbolic trajectories.

The rest of the paper is organized as follows. Section 2 introduces the concept of multi-dimensional trajectory along with a brief overview of symbolic trajectories. Section 3 presents the query processing strategy. The demonstration outlined in Section 4 illustrates a few representative *hybrid* queries. Section 5 concludes the paper.

II. Multi-dimensional trajectories

A. Background:symbolic trajectories

A symbolic trajectory defines a mapping from time to a categorical attribute [2]. Formally, a symbolic trajectory is in its basic form just a time dependent label, that is, a function from time into label values. Labels are just short character strings. Such a function can be represented as a sequence of pairs, called *units*, $< (i_1, l_1), ..., (i_n, l_n) >$ where i_j is a time interval and l_j a label. Time intervals are disjoint (possibly adjacent) and the pairs in the sequence are ordered by time. For example, a simple symbolic trajectory would be:

```
< ([8:30-8:45], walk), ([8:45-9:13], train)..>
```

The model includes a language for pattern matching and rewriting of symbolic trajectories. Matching is used to retrieve symbolic trajectories fulfilling a given pattern. Rewriting can be used to translate a symbolic trajectory into some other form, classify it into certain categories, or retrieve the parts of a symbolic trajectory matching a pattern. A simple pattern over trajectories of transportation means is as follows:

```
* (_ taxi) (_ bus) *
```

In this case the pattern is matched when the transportation mode taxi is followed by bus. Patterns can include variables and conditions over variables, as in the next example where the variable X is bound to the following unit and used within the condition.

```
* (_ taxi) X(_ bus) *
 //duration(X.time) > 20 * minute
```

A pattern appears on the left side of a rule in the rewrite operation, as follows. The outcome is one or more sub-trajectories.

```
* W (monday taxi) X (_ bus) *
// duration(X.time) > 20 * minute
=> W X
```

In contrast with existing works on spatio-temporal pattern matching such as [5], which are exclusively focused on the query language, the symbolic trajectories data model is full integrated into a moving object database, and thus can inter-operate with conventional spatial and trajectory data models.

B. Handling multiple dimensions

Applications may require rich mobility information. For example, for the analysis of the mobility habits in a city, it might be useful not only to know the traces of individuals but also the transportation means they use and possibly the weather conditions during the trips. This information can be obtained from different sources, e.g. weather information services and user's input. Dimensions are thus naturally independent but conceptually refer to the same entity and thus are temporally and spatially related.

Formally, a dimension is a named time-varying function defined over a temporal domain and ranging over a simple domain (numeric, string or spatial). A multi-dimensional trajectory is simply a set of dimensions (*m-trajectory* hereinafter). A m-trajectory has a *temporal extent* defined by the union set of the temporal domains of the dimensions $d_1, .., d_n$. Moreover, at any instant t the m-trajectory has a *value* given by the vector: $< d_1(t), ..., d_n(t) >$ where $d_i(t)$ is the value of the function named d_i at time t. Since the dimensions are not temporally aligned, it may occur that one or more dimensions are $\perp$ (undefined). The set of time intervals in which all of the dimensions are possibly defined, is called *overlap* time.

We consider the class of m-trajectories consisting of one symbolic dimension and one spatial dimension. A symbolic trajectory denotes a sequence of states or events [6]. For the sake of generality, we assume that the units of a symbolic trajectory cannot be splitted or reduced in time, unless compromising the actual meaning of the attribute (i.e. attributes may be *anti-homogeneous* [6]). The spatial dimension varies continuously in time and space. We start illustrating the preliminary operation for the composition of independent spatial and symbolic dimensions in m-trajectories. The m-trajectories are extracted from the GeoLife database. The operations are expressed in the Secondo language.

C. Example dataset

GeoLife (V. 1.2) is a well-known dataset reporting the traces of a group of individuals monitored in Beijing for over three years. Interestingly, GeoLife consists of two distinct datasets. The main dataset contains the GPS tracks of 178 individuals in the form of timestamped point sequences, i.e. $\{(t_i, p_i)\}_{i\in[1,n]}$ where t_i is the timestamp of point p_i. The second dataset contains the temporally annotated sequences of the transportation means used by a subset of 69 individuals, the group we consider in the next. A sequence takes the form $\{(I_i, l_i)\}_{i\in[1,m]}$ where I_i is a time interval and l_i a string in the set: {`walk, bike, car, bus, airplane, other`}, as exemplified in Figure 1. The sequences of GPS points and transportation modes are not temporally aligned.

Start Time	End Time	Transportation Mode
2007/10/19 05:23:15	2007/10/19 05:51:00	taxi
2007/10/19 05:52:18	2007/10/19 09:39:28	walk
2007/10/19 11:18:44	2007/10/19 11:53:40	bike

Fig. 1. GeoLife: temporally annotated sequences of transportation modes

Data organization. We represent the data about this group of people using the moving object data model [4] extended with symbolic trajectories [2]. Specifically, the GPS tracks of every single user, joined together to form a unique sequence (with temporal gaps), are represented by an object of type *m-point*. The sequence of transportation modes for the individual is represented by an object of type *mlabel* (i.e. a symbolic trajectory).

The whole data is stored in the table:

```
geoLife(UserId: integer, GPStrack: mpoint,
  Transport: mlabel)
```

The next operation creates the table *m_geolife* with a column named *Hybrid* which stores the m-trajectory obtained from the spatial and symbolic trajectories of the previous table. The m-trajectory is an object of type *dim* with symbolic dimension s and spatial dimension p.

```
let m_geolife = geolife feed
  projectextend[Id; Hybrid:
    createdim([p: .GPStrack, s: .Transport])]
  consume
```

III. Querying symbolic and spatial dimensions

Now the goal is to extract the sub-trajectories satisfying non-trivial conditions on both the symbolic and spatial dimensions. The approach we are going to describe can be easily extended to continuous numeric dimensions (e.g. moving real). We start considering the following query over the example table (i.e. m_geolife):

Query 1: Retrieve the sub-trajectories where the user covers a distance of more than 10 km first by foot and next by metro.

The query specifies two conditions: one is on the symbolic dimension (i.e. the user first walks and then takes the metro) and one on the spatial dimension (i.e. the distance covered by the user is greater than 10 km.). A sub-trajectory thus satisfies the query if a time period exists, contained in the temporal extent of the trajectory, in which both these conditions are satisfied. Abstractly the problem can be expressed as follows. Consider a m-trajectory M with the usual dimensions p and s and denote with C_s the pattern on the symbolic dimension and with C_p the spatial condition over the spatial dimension. For the sake of generality, we do not make any assumption over the nature of the geometric condition, i.e. simply it is a boolean function that applies to a spatial trajectory and returns true or false. In the example, C_P = `length(M.p)>10` where length is the function returning the length of $M.p$.

The problem is to determine the set of periods I such that it holds: `I={` I_i `| matches (atperiods(M.s,`I_i`),` C_S`) ∧ eval(atperiods(M.p,`I_i`),`C_P`)}` where *matches* and *eval* evaluate the patterns and the spatial condition, respectively, over the dimensions of the input trajectory, while *atperiods* returns the input dimension temporally restricted to the specified period.

Now the question is how to process the query. We can observe: a) it might be unfeasible to determine the periods in which the spatial condition is satisfied over the continuous dimension (e.g. to determine the sub-trajectories of length greater than 10 km); b) the symbolic trajectory consists of atomic units that by design cannot be divided in smaller units.

It follows that existing techniques, such as the *refinement*-based techniques used for the evaluation of mutual topological relationship between moving objects [7] cannot be applied. These considerations suggests an alternative, generic query processing strategy. The idea is to process the symbolic dimension to extract the set of symbolic sub-trajectories satisfying the pattern. In this way the units remain consistent. The temporal extents of these sub-trajectories, e.g. I_i, are next used to temporally restrict the spatial dimension. The obtained spatial sub-trajectories are next matched against the geometric condition C_P. The subset of periods in which the condition C_P is true forms the result I. Note that the two dimensions p and s, when restricted to I_i may be not temporally aligned. The geometric condition is thus evaluated in the time interval in which the spatial dimension is defined.

Global variables. The symbolic and spatial conditions that appear in the query can be tightly interrelated. To better illustrate the problem consider the following query that slightly modifies query 1.

Query 2: Retrieve the sub-trajectories where the user covers a distance of more than 1 km by foot before taking the metro.

It can be noticed that the pattern is the same as in Query 1 (i.e. `'(_ walk)(_ metro)'`) while the spatial condition only applies when the user walks (and not over the whole trajectory as in the previous case). The conditions are thus tightly intertwined. To flexibly represent these ties we use variables. We have already seen that variables can be used locally to the pattern. Now the idea is to define *global variables*, namely variables that are bound to m-trajectories and whose scope is the whole query. A global variable has thus dimensions. For example, Query 2 can be written as follows. Let X be the global variable with dimensions denoted as X_s and X_p. The symbolic condition is defined by a rewriting rule containing the variable X_s while the geometric condition by an expression containing the variable X_p, as follows:

```
C_s: X_s(_ walk) Y_s(_ metro) => X_s Y_s
C_p: length(X_p)>1000
```

X_s is bound to the set $S = \{s_1, .., s_n\}$ of symbolic sub-trajectories matching the pattern, X_p is bound to a set of spatial sub-trajectories $P = \{p_1, ..p_n\}$ where p_i is possibly defined over the temporal extent of s_i. The query is satisfied by the pairs (s_i, p_i) satisfying both conditions. This mechanism is simple, yet, as we will see, quite powerful.

Query operators: We specify two operators for querying m-trajectories consisting of one symbolic dimension and one or more continuous dimensions. The operators are called *h_match* and *h_rewrite*, with h standing for hybrid. Let M be a multi-dimensional trajectory, S the pair: (name$_s$, C_s), and P the list of pairs (name$_{P_i}$, C_{P_i})

- `h_match(M, S, P)` is a boolean function. It returns true if there exists at least one m-trajectory satisfying the conditions in S and P. In this case the symbolic condition in C_S is expressed as a pattern.
- `h_rewrite(M, S, P)` returns the set of m-trajectories satisfying the conditions. In this case the symbolic condition in C_S is expressed as rewriting rule.

IV. Demonstration outline

In the demo we illustrate some examples of complex queries over the GeoLife database. The database is accessed through a Web application interacting with the Secondo system. Upon connection and after being authenticated, the user is presented with the graphical interface offering a set of operations to connect and interrogate a database using SQL or the Secondo query language. An m-trajectory is visualized by combining in an intuitive way the geometric and the thematic information. In particular the spatial dimension is represented by a moving point, i.e. a linear geometric element; the symbolic dimension by graphical attributes, e.g. color, applied to the linear elements.

A. Query examples

Example 1. The query is to retrieve the sub-trajectories where individuals bike for more than 8 km from one place where they stay in the morning to another place where they stay in the afternoon. The staying condition is captured by the transportation mode 'walk'.

The query conditions over dimensions p and s are expressed in textual form as follows. Since we need to extract sub-trajectories the symbolic condition is expressed by a rule. In this case we use three variables X, Y, Z. However only one of them is used in the spatial condition.

```
let C_s = '* X_s[(morning walk)]+
  Y_s[(_ bike)]+ Z_s[(afternoon walk)]+ * //
(Z_s.end - X_s.start ) <
  [const duration value (1 0)]
=> X_s Y_s Z_s'

let C_p = 'length(Y_p, "WGS1984") > 8000'
```

The query over the table m_geolife is as follows. The query stores in the attribute *bike* the resulting sub-trajectories. One of them is displayed in Figure2.

```
query m_geolife feed  extendstream [Bike:
  h_rewrite([.Hybrid, C_p, C_s])]
  project[Id, Bike]
  consume
```

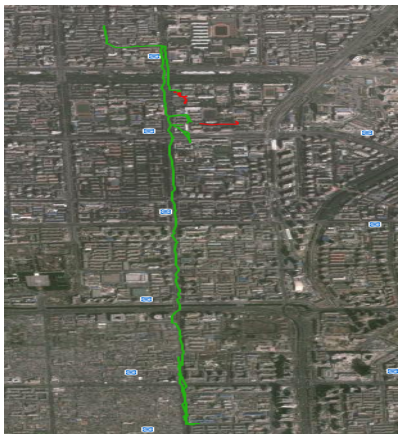

Fig. 2. Example 1: Display of the resulting m-trajectory: the green color indicates bike, the red color walk.

Fig. 3. Example 2: Display of the resulting m-trajectory: the blue lines indicate the public transportation, the red lines the walks. It can be noticed that the walking parts are close to each other.

Example 2. The query is to retrieve the people moving from home to work and viceversa by foot and public transportation (i.e. bus, metro, train). The (informal) query can be more precisely rephrased as follows:

Retrieve the sub-trajectories of the individuals that in the morning go by foot (i.e. walk) to some point where they take a public transportation means, and next in the afternoon they are back near the same point (less than 40 meters) where they took the transportation means in the morning.

The query conditions are:

```
let C_s =
  '* X_s[(morning walk)]+
  Y_s[(_ bus)| (_ metro)| (_ train)]+ *
  P_s[(_ bus)| (_ metro)| (_ train)]+
  Z_s[(afternoon walk)]+ *//
(Z_s.end - X_s.start ) <
  [const duration value (1 0)]
=> X_s Y_s P_s Z_s'

let C_p = 'distance(val(final(X_p)),
  val(initial(Z_p)),
  create_geoid("WGS1984")) < 40'
```

The left side of the rule C_s specifies the pattern. The matching symbolic sub-trajectories are those starting with a walk in the morning followed by the use of one of the public transportation means and ending with a walk in the afternoon of the same day. The spatial condition C_p specifies the distance between the point at which the user gets on the transportation means and the point at which he gets off when back. The distance is measured in meters. The whole query adding the result as new column Loop is as follows, while the extracted sub-trajectory is illustrated in Figure 3.

```
query m_geolife feed
  extendstream[Loop:
    h_rewrite([.Hybrid ,C_s, C_p])]
  project[Id,Loop] consume
```

V. Conclusions

The notion of multi-dimensional trajectory enables a smooth integration of the classical moving object data model with the novel symbolic data model. Future work will focus on the semantic enrichment of symbolic trajectories, on the integration of multiple symbolic dimensions and on the extension of the query processing strategy to account for different classes of spatial conditions and queries.

References

[1] C. Parent, S. Spaccapietra, C. Renso, G. Andrienko, N. Andrienko, V. Bogorny, M.L. Damiani, A. Gkoulalas-Divanis, J. Macedo, N. Pelekis, Y. Theodoridis, and Z. Yan, "Semantic trajectories modeling and analysis," *ACM Computing Surveys*, vol. 45, no. 4, pp. 1–32, 2013.

[2] R. H. Güting, F. Valdes, and M.L. Damiani, "Symbolic Trajectories," FernUniversität in Hagen, Informatik-Report 369 - 12/2013, Tech. Rep., 2013.

[3] F. Valdés, M.L. Damiani, and R.H. Güting, "Symbolic Trajectories in Secondo: Pattern Matching and Rewriting," in *Proc. DASFAA*, 2013.

[4] R. H. Güting, M. Böhlen, M. Erwig, C. Jensen, N. Lorentzos, M. Schneider, and M.Vazirgiannis, "A foundation for representing and querying moving objects," *ACM Trans. Database Systems*, vol. 25, no. 1, pp. 1–42, 2000.

[5] M. R. Vieira, P. Bakalov, and V. J. Tsotras, "Querying trajectories using flexible patterns," in *Proc. EDBT*, 2010.

[6] J. F. Allen and G. Ferguson, "Actions and events in interval temporal logic," University of Rochester, US, Tech. Rep. 521, Tech. Rep., 1994.

[7] M. Schneider, "Evaluation of spatio-temporal predicates on moving objects," in *Proc. ICDE*, 2005.