SeMiTri: A Framework for Semantic Annotation of Heterogeneous Trajectories

Zhixian Yan
EPFL, Switzerland
zhixian.yan@epfl.ch

Dipanjan Chakraborty
IBM Research, India
cdipanjan@in.ibm.com

Christine Parent
University of Lausanne
christine.parent@unil.ch

Stefano Spaccapietra
EPFL, Switzerland
stefano.spaccapietra@epfl.ch

Karl Aberer
EPFL, Switzerland
karl.aberer@epfl.ch

ABSTRACT

GPS devices allow recording the movement track of the moving object they are attached to. This data typically consists of a stream of spatio-temporal (x,y,t) points. For application purposes the stream is transformed into finite subsequences called *trajectories*. Existing knowledge extraction algorithms defined for trajectories mainly assume a specific context (e.g. vehicle movements) or analyze specific parts of a trajectory (e.g. stops), in association with data from chosen geographic sources (e.g. points-of-interest, road networks). We investigate a more comprehensive semantic annotation framework that allows enriching trajectories with any kind of semantic data provided by multiple 3rd party sources.

This paper presents SeMiTri - the framework that enables annotating trajectories for any kind of moving objects. Doing so, the application can benefit from a *"semantic trajectory"* representation of the physical movement. The framework and its algorithms have been designed to work on trajectories with varying data quality and different structures, with the objective of covering abstraction requirements of a wide range of applications. Performance of SeMiTri has been evaluated using many GPS datasets from multiple sources – including both fast moving objects (e.g. cars, trucks) and people's trajectories (e.g. with smartphones). These two kinds of experiments are reported in this paper.

Categories and Subject Descriptors

H.2.8 [**Database Applications**]: [Spatial databases and GIS, Data mining]

General Terms

Algorithms, Design, Experimentation

Keywords

Trajectory Annotation, Semantic Trajectory, SeMiTri

EDBT 2011, March 22–24, 2011, Uppsala, Sweden.

1. INTRODUCTION

GPS chipsets are being embedded in all kinds of moving objects (cars, shipments, smartphones etc.), allowing for large-scale collection of movement data. Such data play an essential role in a variety of well-established application areas (e.g., tracking, urban planning, traffic management, geo social networks). Consequently, research in movement data analytics is blooming and is increasingly focusing on providing rich information services tailored for the application at hand. Most of these services build upon some semantic interpretation of movement data [9][19][15][33]. Therefore, raw data stream (collected via GPS) is first turned into a set of application-meaningful units, called *trajectories*. Trajectories are then enriched with semantic data from the application world. In addition, knowledge extraction techniques such as data mining and analysis of trajectories are applied to summarize the detailed data into manageable information. Extracting semantic behaviors of moving objects is an example of the many open research problems currently being investigated. Many applications indeed are more interested in behavioral aspects than in merely positional data. Trajectory semantics may be inferred from spatio-temporal properties of the raw data stream (e.g. when and where the object stops or moves, its track orientation or movement pattern), from the geographical information related to the region traversed by the trajectory (e.g. its road network, its remarkable features), as well as from application objects stored in the application databases and related to the trajectory (e.g. the list of company customers visited by company salespersons).

We developed a framework, called SeMiTri (Semantic Middleware for Trajectories), whose goal is to support semantic enrichment of trajectories exploiting both the geometric properties of the stream and the background geographic and application data. Semantic enrichment materializes as *annotations* embedded into the trajectory data, i.e. additional data attached to the spatio-temporal positions in the trajectory and encoding extra knowledge about the trajectory. Examples of annotations include recording the observed "activity" of a moving animal (with activity values "feeding", "resting", "moving", etc.), computing and recording the instant speed of the moving object, inferring and recording the "means of transportation" used by a moving person (e.g. by foot, bus, metro, bicycle). The paper presents the semantic model and annotation framework we adopted in SeMiTri for describing and managing the movement of an object.

Our model, semantic annotation algorithms and framework are designed to be generic and applicable on the various kinds of trajectories created by moving objects of different types. SeMiTri is therefore offered as a unified solution to annotate trajectories with semantic information that can readily be exploited by applications to make sense (their own sense) out of movement data.

1.1 Background and Motivation

Data management techniques (data modeling, indexing, querying) for large spatio-temporal data have been investigated during the last decade [11][8]. There is also a large body of work in various techniques for trajectory mining, e.g. pattern discovery, similarity measures, clustering and classification [12][17][13]. These studies focus on raw trajectories, therefore missing the related semantic information contained in the background geographic and application databases.

A new research branch focuses instead on *semantic* analytics of trajectories. The GeoPKDD[1] and MODAP[2] projects [1][26] and other recent works [6] emphasize the need to address semantic behaviors of moving objects. Work in this area focuses primarily on semantic models and trajectory knowledge discovery. Proposed models address different levels of concern, e.g. basic concepts (e.g. data types such as moving point/region [11][23]), conceptual models [26], and ontologies [29]. Work in semantic knowledge mining using geographic data sources mostly focus on specific types and parts of the data stream (cf. details in §2). For example, the activity and point-of-interest mining described in [6][32] consider only trajectory parts where the moving object stopped. On the other hand, map-matching algorithms [4][18][21] focus on mapping vehicle trajectories to roads, considering only the parts where the vehicle is moving. Hence, these works have limited scope for the holistic semantic analysis of the whole trajectory that SeMiTri intends to support.

Applications do benefit from semantic enrichment of trajectories. E.g., when analyzing people trajectories, rather than using the GPS data, we can easily imagine that the application prefers to view a trajectory as the following semantically encoded sequence of triples: *(home, -9am, -)* → *(road, 9am-10am, on-bus)* → *(office, 10am-5pm, work)* → *(road, 5pm-5:30pm, on-metro)* → *(market, 5:30pm-6pm, shopping)* → *(road, 6pm-6:20pm, on-foot)* → *(home, 6:20pm-, -)* (see Fig. 1). Notice that the first and last triples respectively denote the first (Begin) and last (End) spatio-temporal positions delimiting the trajectory. In all triples the spatial ($\langle x, y\rangle$) location is encoded at the semantic level with labels such as "home", "office", "road", "market", expressing the application's interpretation of the location. The second element denotes the time period where the two other elements remain constant (i.e., same location, same annotation). The third element in the triples conveys additional semantic annotation, in this case related to the activity (work, shopping) or to the means of transportation (on bus, on metro, on foot). Clearly, abstracting trajectory data to such a semantic representation enables a better understanding of the semantic behavior. Further, analytics on such semantic trajectories enables contextual and relevant information discovery (e.g. semantic similarity, semantic pattern mining, mobility analysis/statistics), significantly empowering applications.

1.2 Challenges

Designing a generic and efficient annotation framework is non-trivial as many different issues have to be addressed.

(1) The framework should be application-independent while being able to support the specific requirements of any potential applications (e.g. traffic monitoring, semantic location analysis). For example, different levels of granularity are required to analyze movement of: cars between cities or within a city, and people between shops in a commercial center. Car movement is constrained by the underlying road network, while walking follows unplanned paths through places such as parks and buildings. Therefore, no application-specific data should be hard-coded into the framework. Instead, the framework should have the capability to acquire from 3rd party information whatever geographic or application-specific data is needed and input it into its algorithms.

(2) While being generic, the annotation algorithms should exhibit a good performance whatever the characteristics and data qualities of trajectories are. Sampling rates and GPS signal availability influence the quality of raw trajectory data. E.g., while vehicles mostly enjoy good GPS coverage, GPS signal may be lost at people's indoor movement. Trajectories might lack enough data to precisely locate which building the person entered. As a result, mapping trajectories to location artifacts in complex environments such as dense urban areas is a challenge. The algorithms should be able to handle variations in data quality while annotating trajectory.

(3) Providing a holistic annotation framework usually calls for integration of several independent information sources. A priori, the amount of candidate sources for annotation data is high and spatially dense. The framework needs to be able to select the most relevant sources and the most relevant kinds of annotation data for each trajectory segment. For example, it does not make sense to annotate a moving car with the list of restaurants or other location artifacts it quickly passes by, unless it stops around one for certain activity. Overwhelming coverage of space is frequently a problem. For example, a major difficulty in choosing points of interest closest to annotate a given trajectory is not in distance computation but in relevance evaluation. The location where a person stops for shopping in a city center may be associated to many shops in the vicinity. Therefore, we need to infer the exact shop the person stopped for.

(4) For computational efficiency, annotating each GPS point may result in information overload. The trajectory semantic model must offer generic means of semantically aggregating correlated records and provide their condensed representation at the semantic level. To summarize, the challenges we need to address can be stated as:

- To provide a framework that covers the requirements of a wide range of applications. The framework includes both the specification of a generic conceptual model, as well as the specification and implementation of annotation algorithms that exhibit a good performance over a wide range of requirements and data qualities.
- To enable determining which kinds of semantic annotation data should be extracted from available sources and how to appropriately filter it to match the moving object at hand.
- To design efficient annotation algorithms, since the available datasets are large and quickly growing, and annotation data is even required in real-time.

[1] Geographic Privacy-aware Knowledge Discovery and Delivery – http://www.geopkdd.eu/

[2] Mobility, Data Mining, and Privacy – http://www.modap.org/

1.3 Contributions

SeMiTri provides a set of software tools that enable progressively turning raw mobility data into semantic trajectories readily suitable for use by applications. It aims at maximizing annotation relevance while minimizing the computational cost of data annotation. We define a conceptual model (*Semantic Trajectory Model*) that describes a trajectory as a sequence of *semantic episodes* that correspond to an application's interpretation of trajectories. Following a layered approach, we provide algorithms for semantically annotating trajectory episodes with geographic and application data. SeMiTri first exploits latent motion context (e.g. spatio-temporal data) to structure trajectories into stop and move episodes [30]. Next it exploits the geographic context to annotate stops and moves with the geographic objects (be regions, lines and points) that, considering the time period of the stop/move, are relevant to the application. The core contributions described in this paper are:

- A semantic model and a multi-layer framework enabling flexible annotation of trajectories at different levels of trajectory data abstraction.
- Specification and implementation of suitable algorithms for trajectory annotation. This requires novel annotation algorithms exploiting contextual geographic and application information.
- Evaluation of the framework with several vehicle and people trajectories (more have been studied to demonstrate the capability of SeMiTri to work with a variety of datasets of different quality).

2. RELATED WORK

Existing works relevant to our problem are largely piecemeal, diving into algorithms for matching spatial information (semantics in terms of geographic knowledge) to specific type and part of trajectories. Dedicated algorithms are independently designed for trajectory annotations with geographic regions, lines or points. In such case, well segmented trajectories are assumed given in advance as data inputs.

Regarding trajectory annotation with geographic regions, studies focus on computing topological correlations (called *spatial predicates*) between trajectories and regions. For example, Alvares et al. [1] apply spatial joins between trajectories and a given set of regions of interests (ROIs), computing frequent *moves* between *stops* - two important trajectory episodes adopted from the work of Spaccapietra et al.[26]. Other works (e.g., [20]) apply similar algorithms to cloak user locations for preserving privacy.

Regarding trajectory annotation with geographic lines, one significant area is developing efficient *map matching* algorithms to improve matching accuracy with low computing time. Map matching is aiming at identifying the correct road segment on which a vehicle is traveling and even to approximate the vehicle's position on the segment [24][4]. Map matching methods can be best classified into three categories: geometric [3], topological [27], or recent advanced methods [21][18]. Geometric methods use only geometric information of the underlying road network, applying *distance measurements* like point-to-point, point-to-curve (e.g. perpendicular) and curve-to-curve (e.g. Fréchet). Topological methods account for the connectivity and contiguity of the road networks, rather than only the geometric distances. The central focus is to create a better global algorithm for map-matching considering characteristics of vehicle trajectories. The recent methods are designed for handling ambiguous data (noise, sparseness) [21], low sampling rates [18] etc. Traditional map-matching techniques target high matching accuracy, which is usually for movement with unique vehicle (e.g. car or truck). On the contrary, our focus is on additional semantic annotation of the move parts of trajectories, i.e. further inferring transportation mode for each movement episode, which is based on the results of but more than the pure map matching.

Complementary research also focuses on identifying meaningful (significant) points of interests (POI) related to trajectories, based on clustering [35][22] or reinforcement inference techniques (e.g. HITS and PageRank) [6][34]. In addition, [28] designs a semantic spatio-temporal join method to infer activities from trajectories, based on a small set of predefined activity hotspots. [16] mines periodic behaviors in trajectories, focusing on brief semantics like *home/office*.

These prior works focus on situation specific mining and are applicable to annotate only certain types and parts of a trajectory [1][22][28][21], e.g. map-matching for vehicle moves or extracting important POIs for hotspots. None of these studies consider analysis of complete trajectories which naturally contain heterogeneous semantics, e.g. the example of semantic trajectory in §1.1. It is difficult to adapt them for different types of moving objects (e.g. vehicles, people trajectory) crossing geographies of different nature. Moreover, extracting such heterogeneous semantics needs multiple geographic data sources to be combined meaningfully. Our objective is to create a holistic framework for end-to-end annotation of heterogeneous trajectories.

3. SEMITRI APPROACH

We first present the conceptual model of semantic trajectories that we use within SeMiTri, then discuss the annotation principles, and finally present the system architecture.

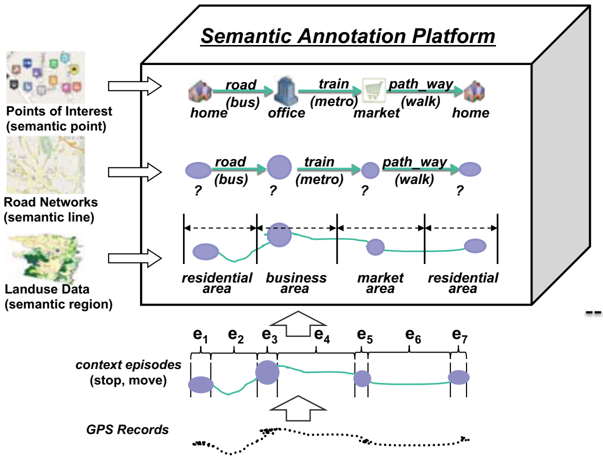
3.1 Model and Definitions

The raw data stream of a moving object, generated by GPS-alike mobile positioning sensors[3], is recorded as a sequence of spatio-temporal points, consisting in *(longitude, latitude, timestamp)* triples. A trajectory identification step [30], not discussed here, splits this sequence into a set of finite subsequences that are meaningful units for the application. These subsequences are called *raw trajectories*.

Definition 1 (Raw trajectory – $\mathcal{T}$). - *A sequence of spatio-temporal points recording the trace of a moving object, i.e.* $\mathcal{T} = \{Q_1, \ldots, Q_m\}$*, where* $Q_i = (x, y, t)$ *is a triple with the positioning* $(longitude\ x, latitude\ y)$ *at timestamp* t.

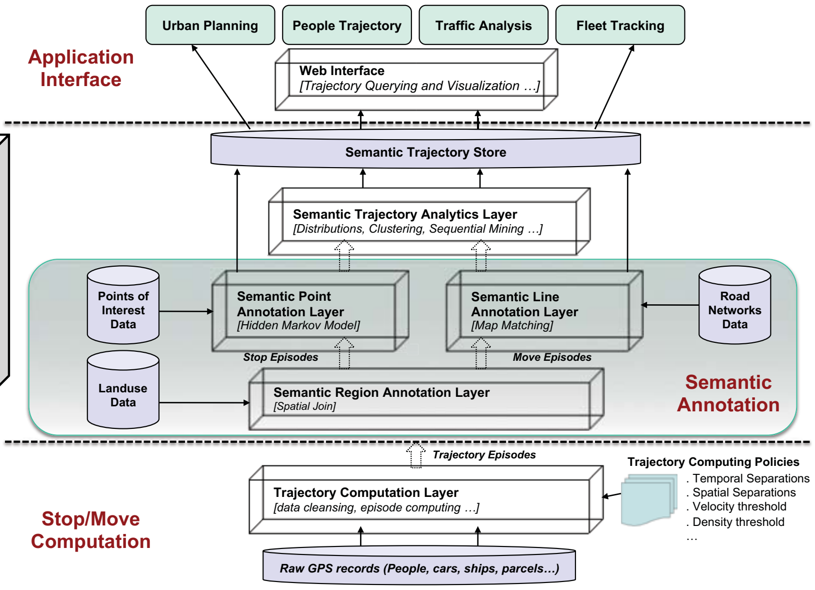
Raw trajectories $\mathcal{T}$ are of varying size, depending on tracking time and location update frequency. There may be gaps in the recording due to several reasons, e.g. signal loss, battery outage, network disconnections, etc. In addition, moving objects or users generating trajectories can use various transportation modes (e.g. walk, metro/bus, bike) which can bring highly varying trajectory characteristics (e.g. acceleration, velocity). We call all of these diverse trajectories *heterogeneous* trajectories – the focus of our annotation task.

[3] Though there are other location-tracking techniques (e.g. triangulation, GSM), our focus is on trajectories in outdoor environments, typically captured using GPS.



Semantic Annotation Platform
Points of Interest (semantic point)
Road Networks (semantic line)
Landuse Data (semantic region)
road (bus)
train (metro)
path_way (walk)
home
office
market
home
?
?
?
?
residential area
business area
market area
residential area
e1 e2 e3 e4 e5 e6 e7
context episodes (stop, move)
GPS Records

Figure 1: Logical View of SeMiTri



Urban Planning
People Trajectory
Traffic Analysis
Fleet Tracking
Application Interface
Web Interface
[Trajectory Querying and Visualization ...]
Semantic Trajectory Store
Semantic Trajectory Analytics Layer
[Distributions, Clustering, Sequential Mining ...]
Points of Interest Data
Semantic Point Annotation Layer
[Hidden Markov Model]
Semantic Line Annotation Layer
[Map Matching]
Road Networks Data
Stop Episodes
Move Episodes
Landuse Data
Semantic Region Annotation Layer
[Spatial Join]
Semantic Annotation
Trajectory Episodes
Trajectory Computing Policies
. Temporal Separations
. Spatial Separations
. Velocity threshold
. Density threshold
...
Stop/Move Computation
Trajectory Computation Layer
[data cleansing, episode computing ...]
Raw GPS records (People, cars, ships, parcels...)

Figure 2: System Architecture of SeMiTri

As a first move towards a semantic representation of trajectories, we introduce *Semantic Places* ($\mathcal{P}$) as the semantic counterpart of the spatio-temporal positions. They denote geographic objects defined in or inferred from 3rd party information sources that contain data about the geographic objects of interest to the application at hand.

Definition 2 (Semantic Places – $\mathcal{P}$). *A set of meaningful geographic objects used for annotating trajectory data. Each semantic place sp_i has an extent and other attributes describing the place. The set ($\mathcal{P}$) is partitioned into three subsets that are defined according to the geometric shape of their extent*[4]*, i.e. $\mathcal{P} = \mathcal{P}_{region} \bigcup \mathcal{P}_{line} \bigcup \mathcal{P}_{point}$, where* (1) *$\mathcal{P}_{region} = \{r_1, r_2, \cdots, r_{n_1}\}$ is a set of places whose extent is a region;* (2) *$\mathcal{P}_{line} = \{l_1, l_2, \cdots, l_{n_2}\}$ is a set of places whose extent is a line; and* (3) *$\mathcal{P}_{point} = \{p_1, p_2, \cdots, p_{n_3}\}$ is a set of places whose extent is a point.*

Due to the spatial extent associated with these geographic objects it is possible to couple a spatio-temporal position in a trajectory with the semantic places whose extent covers this position. Thus we can annotate each spatio-temporal position of a trajectory with links to the semantic place objects that the moving object has (at least we infer so) visited. This is a specific kind of annotation, the geographic reference annotations. Another kind of annotations may also need to be inferred, additional value annotations, which contain extra semantic values, e.g. "work"/"shopping" for activities at stops or "bus"/"walking" for transportation modes in moves. We call *Semantic Trajectory* ($\mathcal{ST}$) a trajectory enriched with annotations of these two kinds.

Definition 3 (Semantic Trajectory). *A trajectory where spatio-temporal positions are complemented with annotations. i.e. $\mathcal{ST} = \{Q'_1, Q'_2, \cdots, Q'_m\}$, where $Q'_i = (x, y, t, \mathcal{A})$ is a tuple defining a spatio-temporal point (x,y,t) and its possibly empty set of associated annotations $\mathcal{A}$.*

Another way of enhancing the knowledge on a trajectory is to identify specific trajectory segments that are semantically meaningful for the application. We call *episode* a maximal sub-sequence of a trajectory such that all its spatio-temporal positions comply with a given predicate that bears on the spatio-temporal positions and/or their annotations. Usually a trajectory is segmented into a list of episodes of several types according to a set of predicates, each predicate defining a type of episodes. For instance, a vehicle trajectory may be segmented (partitioned) into episodes of two types, "stop" and "move", according to the two predicates: (1) for stops: speed$<$ threshold δ, (2) for moves: speed$\geq\delta$. Semantic annotation $\mathcal{A}$ on episode is much more efficient than directly annotating each GPS record in the raw trajectory.

A given trajectory may be structured into episodes in many different ways, i.e. using different sets of predicates. For example, a trajectory of a person within a city may be segmented into episodes based on 1) the means of transportation used, 2) the time periods such as morning, noon, afternoon, evening, and night, and 3) the zones traveled within the city. Each annotation attribute may define its list of episodes e.g. by cutting the trajectory each time the value of the annotation attribute changes. We call each list of episodes an *interpretation of the trajectory*, and a trajectory enhanced with episode annotations a *structured semantic trajectory* ($\mathcal{SST}$).

Definition 4 (Structured Semantic Trajectory). *A representation of a semantic trajectory as a sequence of episodes defined by a set of predicates. $\mathcal{SST} = \{ep_1, ep_2, \cdots, ep_m\}$, such that each episode corresponds to a subsequence of the original trajectory and is represented as a tuple $ep_i = (sp, time_{in}, time_{out}, \mathcal{A})$ where sp is a link to a semantic place ($sp \in \mathcal{P}$), $time_{in}, time_{out}$ are the time the moving object enters and exits sp, and $\mathcal{A}$ is a set of other annotations associated to the whole episode.*

In the sequel of the paper we use the term trajectory (alone) to denote a structured semantic trajectory (Def. 4), and the terms raw trajectory and semantic trajectory to denote non-structured trajectories (Def. 1 and Def. 3).

[4] Region (or area), line and point are standard spatial data types routinely used in GIS. Their formal definition can be found in e.g. [10]. $\mathcal{P}_{region}, \mathcal{P}_{line}, \mathcal{P}_{point}$ are also denoted as *Regions of Interest* (ROI), *Lines of Interest* (LOI), and *Points of Interest* (POI).

3.2 Annotation Design Principles

Fig. 1 illustrates the semantic trajectory computation methodology in SeMiTri. This design is based on the following broad design principles:

1) **Exploit Latent Motion Context**: Context (e.g. whether the object is moving or stationary) is exploited in various ways. First, it plays a role in choosing relevant annotations (e.g. whether to map a trajectory segment to a road or to the nearest restaurant). Second, context persistence supports annotating trajectory episodes rather than annotating each individual GPS point. This obviously saves storage space. Such context can be extracted directly from the raw data stream, based on homogeneous (spatio-temporal) correlations (*density, velocity, direction etc.*) present in the stream [30].
2) **Layered Approach**: SeMiTri follows a layered approach, carefully designed to support efficient semantic annotation. SeMiTri progressively annotates trajectory episodes with *semantic places* $\mathcal{P}$ – first provide a coarse-gained annotation with $\mathcal{P}_{region}$; second provide a fine-gained annotation with $\mathcal{P}_{line}$ and $\mathcal{P}_{point}$, enabling e.g. stop/-move annotation used for later decision making. Section 4 presents the details of all the annotation layers.
3) **Heterogeneity of Semantic Places**: SeMiTri provides algorithms to map trajectory episodes to three categories of geographic objects: ROIs such as *park*, *administrative region* and *landuse cells (residential, industrial)*; LOIs such as *jogging path*, *highway* and other *roads*; and POIs such as *bar*, *restaurant* or even a big *shopping mall*.

3.3 System Architecture

Fig. 2 illustrates SeMiTri's system architecture, showing the various layers and the data flow between the layers. Our system has three main parts, i.e. *Stop/Move Computation*, *Semantic Annotation* and *Application Interface*.

Stop/Move Computation – The raw GPS records are first processed by the *Trajectory Computation Layer*, which performs several data preprocessing operations: (1) remove GPS outliers and smooth the random errors; (2) identify raw trajectories from the initial GPS data stream; (3) segment the raw trajectory into trajectory episodes, based on several computing policies of spatio-temporal co-relations like *density*, *velocity*, *direction* etc.[5] The output trajectory episodes express the motion context (e.g. stop/move). This context can help trajectory annotation in choosing suitable geographic artifacts from 3rd party sources and applying suitable annotation algorithms. For example, the stop episodes need to be annotated with POIs while the move episodes can be integrated with road networks (LOIs).

Semantic Annotation – We design three annotation layers. The *Semantic Region Annotation Layer* receives the stop/move trajectory and uses a state-of-the-art *spatial join* algorithm to pick up *regions* that the trajectory has passed through, primarily to form a coarse-grained view of the trajectory. The *move* episodes are further processed by the *Semantic Line Annotation Layer*. We have developed a new *line annotation* algorithm that is designed to consider heterogeneous trajectories and road networks. Apart from its basic operation of mapping *move* segments to road networks, this algorithm also infers transportation modes exploiting geometric properties/context of the segment (e.g. velocity, acceleration) along with semantic content (e.g. which type of road). The *stop* episodes are funneled to the *Semantic Point Annotation Layer* that computes activity likelihoods and probabilistic estimates of the purpose behind that stop. This is based on a hidden Markov model algorithm that we designed, considering varying spatial densities of possible POIs for heterogeneous (sparse as well as densely populated) geographies. Overlapping ROIs and dense POIs have been traditionally ignored in spatio-temporal data mining [1][28]. The annotations from the three layers are combined to produce the annotated trajectory $\mathcal{SST}$. This "Semantic Annotation" layer is the focus of this paper.

Additional Parts with Application Interface – The computation and annotation result is stored in the *Semantic Trajectory Store*. A *Semantic Trajectory Analytics Layer* encapsulates methodologies that compute statistics about the trajectories (e.g. the distribution of stops/moves, frequent stops, trajectory patterns) and stores them as aggregative information in the store. This data is accessed by applications. We built a *Web Interface* [31] that enables user-friendly queries and visualization of all kinds of trajectories, both semantic and non-semantic ones.

4. ANNOTATION ALGORITHMS

This section explains the details of annotation, considering our objective – *algorithms should exhibit good performance over a wide range of trajectories with varying data quality.*

4.1 Annotation with Semantic Regions

This layer enables annotation of trajectories with meaningful geographic regions. It does so by computing topological correlations of trajectories with 3rd party data sources containing semantic places of spatial kind regions ($\mathcal{P}_{region}$).

The topological correlation is measured using *spatial join* between raw trajectory $\mathcal{Q}$ and semantic regions $\mathcal{P}_{region}$ (i.e. $\mathcal{Q} \bowtie_\theta \mathcal{P}_{region}$). Several forms of spatial predicates are used to compute θ, depending on the type of data. These can be a combination of *directional*, *distance*, and *topological* spatial relations (e.g. *intersection*) [5]. E.g. for *stop* episodes, we found spatial subsumption (ObjectA is *inside* ObjectB) as the most used predicate. For the spatial extent, we use either the spatial *bounding rectangle* of the episode (for move or stop) or its *center* (for stop) to perform spatial join. After finding the appropriate regions (r_i), the layer annotates input trajectories with these regions and associated metadata.

The semantic places are either places computed from the trajectory geometric features (e.g. the bounding box associated to an episode) or identifiable places within some external source. Examples include free form regions like the EPFL campus, a recreation facility with a swimming pool, both taken from Openstreetmap[6], and regions formed from grids of regular cells of repositories such as the Swisstopo[7] landuse and city zones. Fig. 3 shows one person's trajectory on Sunday, annotated with semantic places of various kinds taken from Swisstopo (building area - A, recreational area - B) and Openstreetmap (EPFL campus - C). By using an application database (e.g. EPFL's employee database) annotations for this personal trajectory can be expressed as:

[5] Further details of the trajectory computation operations can be found in [30], though not essential for understanding the semantic annotation algorithms of SeMiTri.

[6] http://www.openstreetmap.org

[7] http://www.swisstopo.admin.ch/

his home → EPFL campus (staying 4 hours) → a swimming pool (staying 1 hour) → his home.

Fig. 4 illustrates landuse classification categories and subcategories that Swisstopo uses to annotate 1,936,439 cells (100m×100m) covering Switzerland.

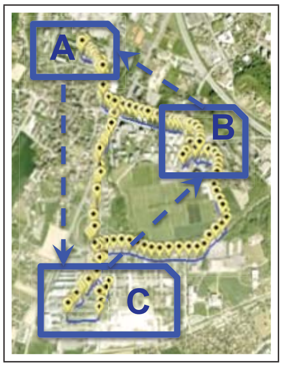
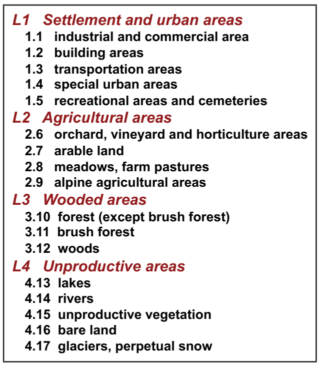


Figure 3: Region Annotation



L1 Settlement and urban areas
1.1 industrial and commercial area
1.2 building areas
1.3 transportation areas
1.4 special urban areas
1.5 recreational areas and cemeteries
L2 Agricultural areas
2.6 orchard, vineyard and horticulture areas
2.7 arable land
2.8 meadows, farm pastures
2.9 alpine agricultural areas
L3 Wooded areas
3.10 forest (except brush forest)
3.11 brush forest
3.12 woods
L4 Unproductive areas
4.13 lakes
4.14 rivers
4.15 unproductive vegetation
4.16 bare land
4.17 glaciers, perpetual snow

Figure 4: Landuse Ontology

Algorithm 1: ***Trajectory annotation with ROIs***

```
Input: (1) a raw trajectory $\mathcal{Q}$ with its sequence of GPS points
       $\{Q_1, \cdots, Q_n\}$, (2) a set of semantic regions
       $\mathcal{P}_{region} = \{region_1, \cdots, region_{n_1}\}$
Output: structured semantic trajectory $\mathcal{T}_{region}$
begin
    $\mathcal{T}_{region} \leftarrow \emptyset$; //initialize the trajectory
    /* compute intersections between $\mathcal{Q}$ and $\mathcal{P}_{region}$; */
    do spatial joins $\mathcal{Q} \bowtie_{intersect} \mathcal{P}_{region}$;
    /* process each intersection and compute trajectory tuple */
    forall intersected regions do
        group continuous GSP point $Q_i \in \mathcal{Q}$ in the intersection;
        approximate entering time $t_{in}$ and leaving time $t_{out}$;
        create a trajectory tuple $\leftarrow (region_j, t_{in}, t_{out}, reg_{type})$;
        if current $reg_{type}$ = previous $reg_{type}$ then
            merge the two tuples into a single tuple; else
            $\mathcal{T}_{region}$.add(tuple); //add the previous tuple to $\mathcal{T}_{region}$;
    $\mathcal{T}_{region}$.add(tuple); //add the last tuple to $\mathcal{T}_{region}$;
    return trajectory $\mathcal{T}_{region}$
end
```

Alg. 1 shows the pseudocode of the annotation algorithm with regions, which directly annotates GPS records with regions. Note that, depending on requirements, the spatial join can be computed only for selected episodes. We apply R*-tree index on semantic regions $\mathcal{P}_{region}$ [2] to improve efficiency of the algorithm. The complexity of the annotation algorithm with region is $O(n * log(m))$, where n is the number of GPS records (or stop episodes) whilst m is the size of $\mathcal{P}_{region}$. For well-divided landuse data, the complexity can be even less, i.e. $O(n)$.

4.2 Annotation with Semantic Lines

This layer annotates trajectories with LOIs and considers variations present in heterogeneous trajectories (e.g. vehicles run on road networks, human trajectories use a combination of transport networks and walk-ways etc). Given data sources of different form of road networks, the purpose is to identify *correct* road segments as well as infer *transportation modes* such as *walking*, *cycling*, *public transportation like metro*. Thus, the algorithms in this layer include two major parts: the first part is designing a global map matching algorithm to identify the correct road segments for the move episodes of the trajectory $\mathcal{Q}$, and the second one is inferring the transportation mode that the moving object used.

Map-matching algorithms usually design a distance metric (e.g. *perpendicular distance*) to map the GPS points to the nearest road segment [24]. Though suitable for well-defined high-way networks, perpendicular distance is not suitable for dense networks, parallel road-ways and arbitrary crossings. This is because vertical projections of (x,y,t) points on corresponding road segments often do not fall on the segment. Thus, we apply the *point-segment distance*, defined as:

$$d(Q, A_iA_j) = \begin{cases} d(QQ') & \text{if } Q' \in A_iA_j \\ min\{d(QA_i), d(QA_j)\} & \text{otherwise} \end{cases} \tag{1}$$

where Q' is the projection of the GPS point Q on the line determined by the two crossings A_i and A_j; $d(QQ')$ is the perpendicular distance between Q and that line; $d(QA)$ is the Euclidean distance between Q and the crossing A.

As a subsequence of raw trajectory $\mathcal{Q}$, a *move* episode also includes a list of spatio-temporal points. Choosing the candidate road segment for each single point independently sometimes results in incorrect mapping, specially for non-perpendicular path ways. Global map matching algorithms have shown better matching quality [4][24] as they consider the context of neighboring points. We adopt this with the *point-segment distance*, in terms of designing two metrics (*localScore* and *globalScore*) to map *move* episodes to appropriate road segments for heterogeneous road structures.

We consider a *global view radius* R around candidate points, with a context window of size $2R$. Therefore, mapping results of point Q depend also on the effects of its neighboring points (N_1 points before and N_2 points after in radius R). For computational efficiency, only the *neighboring* segments are considered as candidate road segments $candidateSegs(Q)$. They can be efficiently accessed with R*-tree index [2]. We normalize the point-segment distance $d(Q, A_iA_j)$ as the $localScore$ between point Q and road segment A_iA_j.

$$localScore(Q, A_iA_j) = \begin{cases} \frac{d_{min}(Q)}{d(Q, A_iA_j)} & A_iA_j \in candidateSegs(Q) \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

where $d_{min}(Q)$ is the shortest distance from Q to all possible candidate road segments A_iA_j. Based on *localScore*, we compute a global measurement - *globalScore* - between Q and A_iA_j considering the context window $2R$ containing N_1 points prior to Q and the forthcoming N_2 points.

$$globalScore(Q, A_iA_j) = \frac{\sum_{k=-N_1}^{N_2} w_k \cdot localScore(Q_k, A_iA_j)}{\sum_{k=-N_1}^{N_2} w_k} \tag{3}$$

$$w_k = \begin{cases} exp(-\frac{d(Q_0Q_k)^2}{2\sigma^2}) & d(Q_0Q_k) < R \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

where Q_k is the k^{th} neighboring point of Q (e.g. Q_0 is Q itself, Q_{-1} is the previous point whilst Q_{+1} is the next point); w_k is the corresponding weight determined by a Kernel smoothing function with the Kernel bandwidth σ.

After the first step of the global map matching, each episode is annotated in terms of a list of road segments, i.e. $ep = \{r_1, r_2, \ldots, r_l\}$. We further infer the annotation of transportation mode on each segment (or route), getting the pairs of $\langle r_i, mode_i\rangle$. In our experiment, we did consider four types of transportation modes, i.e. *walking. bicycle*, *bus* and *metro*. Such annotation is determined by the characteristics of the move episode and the matched road segments, including *average velocity*, *average acceleration*, *road type* etc.

Alg. 2 shows the detailed procedure of the semantic line annotation algorithm: (1) select candidate road segments, (2) calculate the point-segment distance, (3) normalize the distance as *localScore*, (4) compute the weight and calculate *globalScore*, (5) determine the map matching segment for each point based on *globalScore*, (6) further infer the transport mode based on the features of the GPS points on the segment and the road type information.

Algorithm 2: *Trajectory annotation with LOIs*

```
Input: (1) a move episode of raw trajectory $\mathcal{Q}$ of GPS points
       $\{Q_i(x_i, y_i, t_i)\}$
       (2) a set of road segments $P_{line} = \{r_1, r_2, \cdots, r_m\}$
Output: semantic trajectory $\mathcal{T}_{line}$
begin
    $preSeg \leftarrow \emptyset$, $\mathcal{T}_{line} \leftarrow \emptyset$; //initialize the trajectory
    forall $Q_i = (x, y, t) \in \mathcal{Q}$ do
        /* select candidate roads for $Q_i$ (R*-tree)*/
        $candidateSegs(Q_i) \leftarrow \{r_1^{(i)}, \cdots, r_n^{(i)}\}$; // select only
        neighboring road segments
        /* calculate dist., normalize it as localScore */
        compute the distance between point $Q_i$ and
        $\forall r_j^{(i)} \in candidateSegs(Q_i)$;
        choose the closest segment $min\{d(Q_i, r_j^{(i)})\}$ (Equ. 1);
        normalize distance as $localScore(Q_i, r_j^{(i)})$
        $\forall r_j^{(i)} \in candidateSegs(Q_i)$ by Equation 2;
        /* calculate globalScore: (point, segment) */
        choose global points $(Q_{-N_1}, \cdots, Q_{+N_2})$ in radius $R$;
        compute their Kernel smoothing weights by Equation 4;
        compute the $globalScore(Q_i, r_j^{(i)})$ for
        $\forall r_j^{(i)} \in candidateSegs(Q_i)$ by Equation 3;
        /* compute $Q'$ with road position (if needed) */
        rank the computed $globalScore(Q_i, r)$
        choose the highest score to match $segmentId$ for $Q_i$;
        compute the corrected position $(x', y')$ if needed ;
        /* add road segment as a trajectory tuple */
        if $preSeg \neq null$ and $preSeg \neq segmentId$ then
            /* infer transportation mode */
            get $tranportMode$ by velocity distribution, road
            information etc.
            /* add the semantic episode */
            $(segmentId, time_{in}, time_{out}, mode) \rightarrow \mathcal{T}_{line}$;
            $preSeg \leftarrow segmentId$;
    return structured semantic trajectory $\mathcal{T}_{line}$
end
```

Since each GPS point considers only the neighboring road segments as a set of candidate segments (by R*-tree), the candidate set size is significantly smaller than the total size of road networks in real-life datasets. This makes the algorithm, besides having better matching quality, also efficient, with linear complexity on the size of the GPS points $O(n)$. The global map matching parameters (e.g. radius R and kernel width σ) are tuned in the experiment.

4.3 Annotation with Semantic Points

This layer annotates the *stop* episodes of a trajectory with information about suitable *points of interest* (POIs). Examples of POI are *restaurant*, *bar*, *shops*, *movie theater* etc. For scarcely populated landscapes, it is relatively trivial to identify the objective of a stop (e.g. petrol pump on a high-way or only own home in a residential area). However, densely populated urban areas have several candidate POIs for a stop. Further, low GPS sampling rate due to battery outage and signal losses makes the problem more intricate.

We have designed a *Hidden Markov Model* (*HMM*) based technique for semantic annotation of *stops*. Unlike most other algorithms to identify POIs [1][28], an unique novelty of our approach is that it works for densely populated area with many possible POI candidates for annotation, thus catering to heterogeneous people and vehicle trajectories. It also enables identifying the activity (behavior) behind the stop, thus annotating the trajectory with such information.

HMM is a classical statistical signal model in which the system being modeled is assumed to be a Markov process with unobserved state [25]. We consider the *temporal sequence* of GPS stops: $\mathcal{S} = (S_1, S_2, \cdots, S_n)$ as the observed values. Dense urban areas can have several different POIs. E.g. Milan dataset in our experiments has 39,772 POIs with largely varying density. Such large number makes it probabilistically intractable to infer the exact POI from imprecise location records. However, the number of *types* (or categories) of POIs usually is tractable. E.g. Milan POI dataset has five top-categories, i.e. *services, feedings, item sale, person life,* and *unknown.* POI categories add significant semantic content to the stop for activity inference (e.g. Sally stopped for *lunch*), which becomes a tractable problem.

Fig. 5 expresses the resultant HMM problem. The initial input is the raw trajectory $\mathcal{Q}$, i.e. the sequence of (x,y,t) points; a *sequence of stops* is computed and forms the real observation (O); the *exact* POI data are the superficial hidden states, whilst the POI *categories* are the real hidden states that we are interested in. Our goal is to identify the real hidden states and use them to annotate the stops.

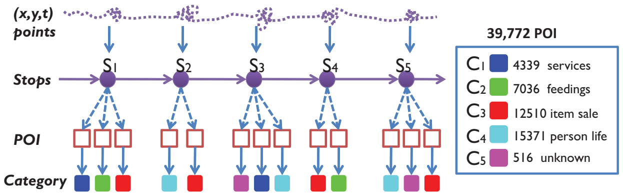


Figure 5: HMM formalism for inferring POI category

$$\mathcal{A} = \begin{pmatrix} 0.8 & 0.05 & 0.05 & 0.05 & 0.05 \\ 0.05 & 0.8 & 0.05 & 0.05 & 0.05 \\ 0.05 & 0.05 & 0.8 & 0.05 & 0.05 \\ 0.05 & 0.05 & 0.05 & 0.8 & 0.05 \\ 0.15 & 0.15 & 0.15 & 0.15 & 0.4 \end{pmatrix}$$

Figure 6: Example state transition matrix

Modeling: Let there be n POI categories $C_1 \ldots C_m$. Typically, a HMM λ has three major components, i.e. $\lambda = (\pi, \mathcal{A}, \mathcal{B})$; where π is the probability of the initial states, i.e. $Pr(C_i)$, $\mathcal{A}$ is the state transition probability matrix $([Pr(C_j|C_i)]_{m\times m})$, $\mathcal{B}$ is the observation probability for each state $Pr(o|C_i)$.

- **Initial Probabilities** (π). We approximate the probability of initial states π as the percentage of POI samples belonging to each category from the information source. Therefore, for Milan POI dataset, $\pi = \{\frac{4339}{39772}, \frac{7036}{39772}, \frac{12510}{39772}, \frac{15371}{39772}, \frac{516}{39772}\}$.
- **State Transition** ($\mathcal{A}$). State transition probability $Pr(C_j|C_i)$ in our formulation represents the possible stop (activity) sequence of user; i.e. probability to perform activity in places belonging to category C_j given his prior activity in places belonging to category C_i. Wherever available, activity sequences (e.g. *home→ work → shop or swim→ home*) are obtained

through other information sources (e.g. from *region* transitions). For trajectories having insufficient history, we initialize the state transition matrix following nomenclatures of the POI categories and object type (e.g. associate high probability for meaningful state transitions and low probabilities for non-meaningful state transitions in Fig. 6). *Learning* dynamic and personalized transition matrix $\mathcal{A}$ is interesting but not the focus of this paper.

- **Observation Probabilities ($\mathcal{B}$).** $Pr(o|C_i)$ intuitively represents the probability of seeing a *stop* o (as the observation) in $\mathcal{T}$ *caused by* user's interest in places belonging to category C_i. $Pr(o|C_i)$ can be approximated by the center of the *stop* $Pr(center_{xy}|C_i)$ or the bounding rectangle $Pr(boundRectangle|C_i)$.

Computing $\mathcal{B}$ for areas having high POI density is not easy. Our solution is based on the intuition that influence of a POI category on a *stop* is proportional to the number of exact POIs of that category in the stop area. We model the influence of a POI as a two-dimensional Gaussian distribution - the mean is the POI's physical position (x, y) and the variance is $[\sigma_c^2, 0; 0, \sigma_c^2]$, where σ_c is category specific. Fig. 7 displays an example of 12 POIs' Gaussian distributions with the corresponding densities in Fig. 8. By Bayesian rule, we deduce the lemma to determine $Pr(o|C_i)$ in $\mathcal{B}$.

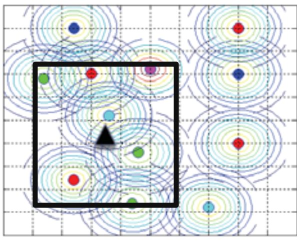


Figure 7: POIs/Discretization

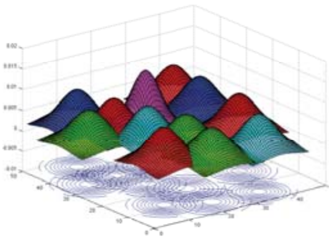


Figure 8: POI densities

LEMMA 1. *$Pr(o|C_i)$ is proportional to the sum of the probability of each POI that belongs to this category C_i, namely $Pr(o|C_i) \propto \Sigma_j Pr(o|poi_j^{(C_i)})$.*

PROOF. of Lemma 1

$$\begin{aligned} Pr(o|C_i) &= \frac{Pr(o, C_i)}{Pr(C_i)} = \frac{\Sigma_j Pr(o, poi_j^{(C_i)})}{\Sigma_j Pr(poi_j^{(C_i)})} \\ &= \frac{\Sigma_j Pr(o|poi_j^{(C_i)}) Pr(poi_j^{(C_i)})}{\Sigma_j Pr(poi_j^{(C_i)})} \\ &\propto \Sigma_j Pr(o|poi_j^{(C_i)}) Pr(poi_j^{(C_i)}) \\ &\propto \Sigma_j Pr(o|poi_j^{(C_i)}) \end{aligned}$$

□

We employ *discretization* and *neighboring* techniques to improve the efficiency of computing $Pr(o|C_i)$. Using *discretization*, we divide the area of POIs into grids (jk) and pre-compute discretized probability values of $Pr(grid_{jk}|C_i)$, as the approximation of $Pr(center_{xy}|C_i)$. Further, for each $grid_{jk}$, we consider only *neighboring* POIs in that box (black rectangle in Fig. 7), instead of all the POIs in the area.

Inferring Hidden States: Using the above defined complete form HMM $\lambda = (\pi, \mathcal{A}, \mathcal{B})$, we infer their hidden states (the purpose behind the stops) $HS = \{pc_1, pc_2, \cdots, pc_n\}$ from the stop sequence $OV = \{stop_1, stop_2, \cdots, stop_n\}$ available through the stop/move computation; where pc_t is the POI category $pc_t \in \{C_1, \cdots, C_m\}$. This problem can be formalized as maximizing the likelihood $Pr(HS|OV, \lambda)$.

We redefine this problem as a *dynamic programming* problem, defining $\delta_t(i)$ as the highest probability of the t^{th} *stop* caused due to POI category C_i (Equation 5). Equation 6 gives the corresponding induced form of highest probability at the $(t+1)^{th}$ stop for category C_j, considering the HMM state transition probabilities. We record the previous state C_i that gives the highest probability to current state C_j by $\psi_{t+1}(j)$ (Equation 7).

$$\delta_t(i) = \max_i Pr(pc_1, \cdots, pc_t = C_i, o_1, \cdots, o_t|\lambda) \quad (5)$$

$$\delta_{t+1}(j) = \max_i\{\delta_t(i)A_{ij}\} \times B_j(o_{t+1}) \quad (6)$$

$$\psi_{t+1}(j) = \underset{i}{\text{argmax}} \quad \delta_t(i)A_{ij} \quad (7)$$

Finally, we employ the Viterbi algorithm [7] to solve this dynamic programming problem for inferring the hidden state (stop category) sequence. We first recursively compute $\delta_t(i)$, and deduce the final stop state with the highest probability in the last stop, then backtrack to the previous stop state by $pc_{t-1}^* = \psi_t(pc_t^*)$. The details of the algorithm for inferring hidden stop category sequence is described in Algorithm 3. The output of this layer is a sequence of semantic episodes describing the stops.

Algorithm 3: ***Trajectory annotation with POIs***

```
Input: (1) an observation sequence of stops
    $O = \{Stop_1, Stop_2, \cdots, Stop_n\}$; (2) points of interest
    $POIs = \{\langle p_1, q_1\rangle, \cdots, \langle p_k, q_k\rangle\}$ where $q_i \in \{C_1, \cdots, C_5\}$
Output: a hidden state sequence about stop behaviors (in terms
    of POI categories), i.e.
    $S = \{q_1, q_2, \cdots, q_n\}, q_i \in \{C_1, \cdots, C_5\}$
begin
    /* learn the model from POIs */
    $\lambda = (\pi, \mathcal{A}, \mathcal{B})$
    /* initialization */
    forall POI category $C_i$ do
        $\delta_1(i) = \pi_i B_i(o_1), 1 \le i \le N$; $\psi_1(i) = 0$
    /* recursion */
    forall t: 2 to n do
        forall categories $C_j$ do
            $\delta_t(j) = \max_i[\delta_{t-1}(i)A_{ij}] \times B_j(o_t)$
            $\psi_t(j) = \text{argmax}_i[\delta_{t-1}(i)A_{ij}]$
    /* termination */
    $P^* = \max_i[\delta_T(i)]$; $q_n^* = \text{argmax}_i[\delta_T(i)]$
    /* state sequence backtracking */
    forall t: n to 2 do
        $q_{t-1}^* = \psi_t(q_t^*)$
    /* get the semantic trajectory with POI tags */
    $S = \{\langle stop_1, q_1\rangle, \cdots, \langle stop_n, q_n\rangle\}$
    summarize $\mathcal{T}_{point}$ from extracted POI sequence
    ($\langle stop, t_{in}, t_{out}, tagList\rangle$).
    return structured semantic trajectory $\mathcal{T}_{point}$
end
```

5. EXPERIMENT ANALYSIS

We implemented SeMiTri and carried out extensive experiments to annotate large GPS trajectories of heterogeneous moving objects with varying data qualities – private cars, taxis and GPS embedded smartphones carried by people.

5.1 Implementation Setup

We implemented and deployed SeMiTri on a Linux operating system - Ubuntu 9.10, with the Intel(R) 2×3.00GHz CPU and 7.9GiB memory. The context computation and semantic annotation algorithms are implemented in Java 6; PostgreSQL 8.4 with spatial extension PostGIS 1.5.1 is used for implementing the different database stores. The raw GPS records and geographic information from 3rd party sources are loaded into the databases in different tables and queried by different layers during execution time (Fig. 2).

The main output of SeMiTri – the *structure semantic trajectories* (Def. 4) is stored in the *Semantic Trajectory Store*. Dedicated tables are designed for *GPS records*, *trajectories*, *stops/moves*, and *annotations with geographic data*, with some new datatypes we defined in Postgis. This is expected to be queried by several trajectory applications. In our experiments, since the datasets are huge, we highlight large-scale aggregated results through the *Semantic Trajectory Analytics Layer*. This layer computes additional statistical information on the trajectories at all the different abstraction levels. In addition, we have developed a *Web Interface* as a pilot application using Apache 2.2.12 Web Server and Tomcat 6.0.26 Application Server. This provides trajectory querying and visualization services to users/applications [31].

Our experiments focus on vehicle (taxi, private cars) and people trajectories. While transportation mode of vehicles is trivial (vehicle type), people can take different choices in several or even one trajectory. Note that trajectories might have varying semantic annotations due to varying amount of 3rd party sources available. For some scenarios, SeMitri produces partial annotations. Our objective here is to bring out the *individual* performance of each layer, which results in the collective annotation of the overall trajectory. To do this, we present annotation performance of each layer with different sources tuned to test the layers individually.

5.2 Vehicle Trajectories

The dataset of vehicle trajectories is shown in Table 1.

- **Trajectories**: We consider (1) 3 millions GPS records of two Lausanne taxis, collected over 5 months by Swisscom[8]; (2) 2 millions GPS records of 17,241 private cars tracked in Milan during one week from the GeoPKDD project; (3) a GPS trace of 2-hour drive of a private car in Seattle, provided by Krumm[9] for testing map matching in line annotation.
- **Place Data Sources**: We use (1) the landuse data of Switzerland on the taxi data to validate the *Semantic Region Annotation Layer*; (2) a large POI dataset of Milan on the Milan private car data for the *Semantic Point Annotation Layer*; (3) the benchmark dataset containing the road network of Seattle as well as the ground truth path for validating and tuning the *Semantic Line Annotation Layer*.

The *Trajectory Computation Layer* produces 172 daily trajectories with 1,824 *moves* and 1,786 *stops* over the Lausanne taxi data. Based on this, the *Semantic Region Annotation Layer* annotates the raw trajectories and the trajectory episodes (stops/moves) with the landuse data. Fig. 9 shows the detailed landuse category distribution over taxi trajectories. Landuse has 4 large categories and 17 sub-categories (from 1.1 to 4.17, see Fig. 4). We observe that most of the taxi GPS records are in *building areas (1.2)* (46.6%) and *transportation areas (1.3)* (36.1%), nearly 83% GPS points belonging to these two categories (see the trajectory column in Fig. 9). The *move* part covers 79.25% of the taxi landuse area, whilst the *stop* part only covers 20.75%. Due to the high-level abstraction into region-based movements, the resultant semantic trajectory $\mathcal{T}_{region}$ representation achieves almost 99.7% storage compression (e.g. 3M GPS records can be annotated with only 8,385 cells).

Map-matching (in *Semantic Line Annotation layer*) is applied on the *move* episodes of trajectories in our experiments wherever road network data is available (for vehicle and people trajectories). To measure the efficiency of our approach, we perform a sensitivity analysis of the algorithm using Krumm's benchmark dataset. We first tune the global view radius (R) and the kernel width (σ) for the input data source. Fig. 10 shows the effect of different σ and R on matching accuracy. We observe that small values of R (=2) and σ(=0.5R) produce very high matching accuracy, similar to the recent results on this dataset [21], confirming the efficiency of the algorithm in fast computation. Nevertheless, the focus of our *Semantic Line Annotation* is not only on the map matching accuracy, but also on the determination of transportation modes in heterogeneous trajectories. This is illustrated through people trajectories in §5.3.

We analyze the performance of the HMM-based *Semantic Point Annotation* algorithm using the POI data in Milan. The 39,772 POIs are divided into 5 categories: 4,339 *services*, 7,036 *feedings*, 12,510 *item sale*, 15,371 *person life* and 516 *unknown* (Fig. 11 - first column). The 3rd party sources have a high density of POIs in this area. Traditional one-to-one match methods like [28] are not suitable here. Our *Semantic Point Annotation* layer enriches the *stop*s computed from the GPS tracks of the private cars and extracts the most probable POI category for each stop. In Fig. 11 (second column), we observe most of the stops (about 56.3%) belong to *item sale* (shopping, groceries etc.) with the next one being *person life* (e.g. sport) (about 24.2%), which makes intuitive sense for private cars trajectories. Through well-defined rules, SeMiTri is able to perform analytics over the extracted semantic trajectories. For example, Fig. 11 (third column) also shows the *trajectory category*, which is defined

Table 1: Datasets of Vehicle Trajectories

	Dataset	*# objects*	*# GPS records*	*Tracking time*	*Sampling frequency*
(1)	Lausanne taxis	2	3,064,248	5 months	1 second
(2)	Milan private cars	17,241	2,075,213	1 week	avg. 40 seconds
(3)	Seattle drive	1	7,531	2 hours	1 second

semantic places (3rd party geographic sources)	(1) Lausanne (Switzerland): landuse - 1,936,439 cells (2) Milan: points of interest - 39,772 POIs (3) Seattle network (Krumm's benchmark): 158,167 road lines

[8] http://www.swisscom.ch/

[9] http://research.microsoft.com/en-us/um/people/jckrumm/MapMatchingData/data.htm

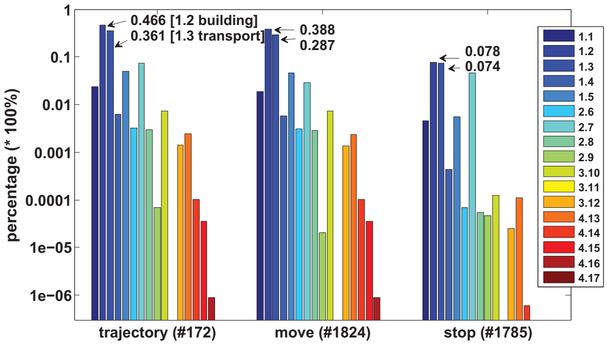


Figure 9: Landuse category distribution for Taxi data (trajectory, moves, stops)

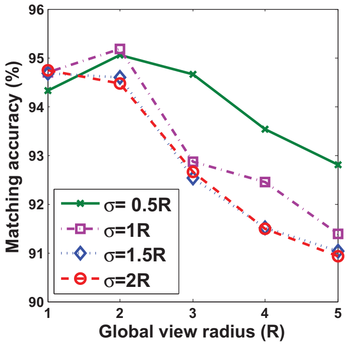


Figure 10: Sensitivity of map matching accuracy w.r.t. R/σ

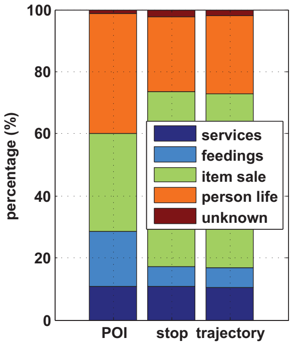


Figure 11: Semantic stops/ trajectories by point annotation

as: *the category of $\mathcal{T}$ is the category which has the maximum stop time* (Equation 8). This can be considered as a semantic classification of the raw trajectories.

$$trajectory_{cat} = \underset{C_i}{\operatorname{argmax}} \sum_{stop.cat=C_i} (stop.time_{out} - stop.time_{in}) \tag{8}$$

Note that the distribution of *trajectory* categories is statistically similar to the distribution of *stop* categories (see Fig. 11). This is because the dataset has only 1.7 stops per trajectory on the average – 2M GPS records of 77,694 trajectories have 133,556 stops, thereby resulting in a similar distribution. This is co-incidental and depends largely on the trajectory data in applications.

5.3 People Trajectories

The dataset of people trajectories is shown in Table 2.

- **Trajectories**: This dataset [14] is provided by Nokia Research Center, Lausanne. They distributed smartphones (Nokia N95) to students/researchers in Lausanne, collecting several *people sensing* data including GPS feeds. We analyzed 185 users, with 23,188 daily trajectories with 7.3M GPS records from this data. Table 2 also describes the details of 1,077 daily trajectories of 6 specific users we know.
- **Place Data Sources**: We use the swiss landuse data.We also extract additional geographic data from Openstreetmap – a publicly available and free editable map site that includes regions, POIs, road networks of several types, and load them into our PostGIS data store (using Osm2pgsql [10]).

People trajectories are far more non-homogeneous than vehicle trajectories: (1) Many phenomenon can result in GPS data loss, such as the limited power of smartphones, battery outage, and indoor signal loss. (2) Non-stationary sampling rates due to on-chip power saving software modules that monitor the sensor; (3) Compared to vehicles, users in people trajectory can take complicated on-road/off-road routes, and choose diverse transportation modes (e.g. *walk*, *bicycle*, *bus*, *metro*) during their daily movements. Therefore, capabilities of SeMiTri are well established through systematic semantic enrichment of such trajectories.

Through trajectory episode (stop/move) computation, the 7.3M GPS records are abstracted as 46,958 moves and 52,497 stops in 23,188 daily trajectories. To understand the type of knowledge inference possible, Fig. 12 shows the loglog plot of the length (i.e. the number of GPS records) of extracted trajectories, stops and moves. It shows that most of *moves/trajectories* have similar patterns, with a large number of GPS records (say more than 10^3), whilst the number of GPS records in *stops* largely stay between 10^2 and 500, decreases from 10^2 to 10^1, and has few unusual cases in $[500, 10^3]$. In addition, Fig. 13 shows the details of *stops* and *moves* for the selected 1,077 daily trajectories of 6 users. Note that the number of GPS records for each user in Fig. 13 is divided by 100, for better representation purposes, and to bring out the storage compression achievement.

Through results of the semantic region annotation layer, we observe a large bulk of stops and moves occur in the *building areas (1.2)* (33.3%) and the *transportation areas (1.3)* (28.6%). However, while 83% of stops/moves in taxi trajectories (Fig. 9) are in these areas combined, only 61% of people trajectories fall in these. This is likely and intuitive, showing people trajectories have much more variations in their movements and areas covered. To discover further insights, Fig. 14 shows the precise distribution of the six people selected (with the list of top-5 categories for each user). We observe, *user*3 has much higher percentage of location records in *lake area (3.12)* because his accommodation is close to the Geneva lake; *user*4's house is in *commercial center area (1.1)*; and *user*2 does a lot of hiking and skiing in *wooded areas (3.10)* - different from other users.

Apart from performing map-matching, people trajectories in *Semantic Line Annotation* are enriched to determine the transportation mode (e.g. *metro, bus, walk etc*). Our *Semantic Line Annotation Layer* considers the underlying network information along with the velocity/acceleration distribution for each road segment from the initial map-matching results to determine the transportation mode. For example, Fig. 15 shows a typical *home-office* trip of *user*4, who walked a few blocks from home, then took the Metro line, and finally walked from the Metro stop to his office: sub-figure (a) shows the original GPS points; (b) displays the initial map-matched road segments for these GPS points; (c) further infers the corresponding different transportation modes such as *metro* or *walk*; finally (d) summarizes the trajectory in terms of meaningful road sequences stored in the semantic trajectory store. In addition to taking the *metro* as shown in Fig. 15, *user*4 has taken three other transportation modes: *bus, bike* or *walk.* In Fig. 16, the left subfigure (a) shows an example of using *bike* for going home to office; whilst in sub-

[10] http://wiki.openstreetmap.org/wiki/Osm2pgsql

Table 2: People Trajectory Data from Mobile Phones

All dataset	user-id	*from-date*	*to-date*	*#days-with-gps*	*#GPS*	*semantic data*
185 smartphone users	1	2009-02-17	2010-04-27	191	50,274	*landuse:*
23,188 daily trajectories	2	2009-02-25	2010-05-16	330	200,418	1,936,439 cells
7,306,044 GPS records	3	2009-09-14	2010-05-16	166	62,272	*swiss-map:*
from date: 2009-02-01	4	2009-11-19	2010-05-16	161	66,304	109,954 points
to date: 2010-08-16	5	2009-12-18	2010-05-16	140	69,467	344,975 lines
	6	2010-01-25	2010-05-16	89	45,137	233,896 regions

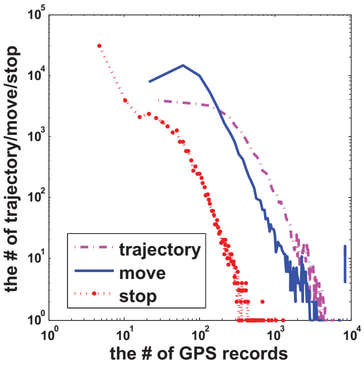


Figure 12: Trajectory context computation (distribution)

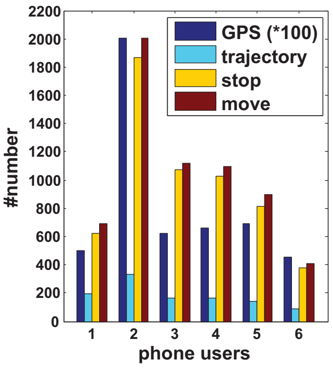


Figure 13: Trajectory context computation (sample)

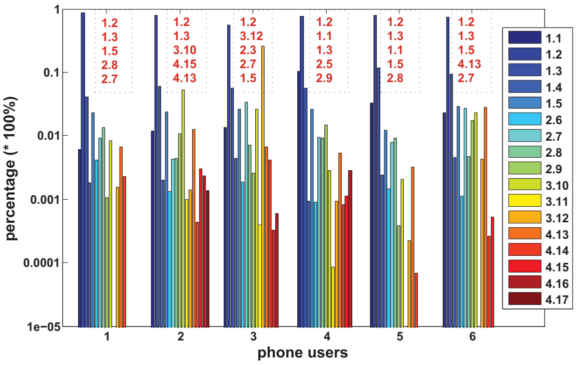


Figure 14: Landuse category distribution and top-5 categories of people trajectories

figure (b) the user took the *bus*, with *walking* as well during the beginning and ending parts of the home-office move, for getting on/off the buses.

Our POI dataset of Lausanne area is sparse at the moment, and does not reflect the real-life POI density of the area (compared with the much more complete Milan city data). We will apply the semantic point annotation method when more POIs with well-defined categories are available. We are currently building such dataset for EPFL campus.

5.4 Discussion

The above experiments validate the semantic annotation part of SeMiTri – ability to annotate *heterogeneous* trajectories comprehensively, creating structured semantic trajectories, by exploiting available 3rd party geographic information sources.

In [31], we developed a *Web Interface* for query and visualization of the annotation results from SeMiTri. This enables users to easily query their raw GPS traces, trajectory episodes, as well as semantic trajectories through a web browser with Google Earth Plugins. Fig. 15 and Fig. 16 also exhibits some examples of such trajectory visualization (KML files), that are retrieved from the trajectory stores.

Finally, we analyzed the run-time performance of SeMiTri. Fig. 17 summarizes the latency distribution of SeMiTri for processing phone trajectories. We observe that computation and annotation latencies are much lower (both *map-matching* and *landuse*) than the storing time (write the results into our semantic trajectory store). For all the six users, the average time for *computing episodes*, *storing episodes*, *map matching annotation*, *storing matched results*, *landuse annotation* for a daily trajectory are respectively 0.008, 3.959, 0.162, 0.292, and 0.088 seconds. Latency distributions for vehicle trajectories are also similar.

6. CONCLUSION

This paper presented SeMiTri's multi-tiered approach towards semantic enrichment of raw trajectories, exploiting context in the stream and geographic data from 3rd party sources. It can significantly enrich trajectory semantics via annotation algorithms.

SeMiTri is designed to work on *heterogeneous* trajectories, different types of moving objects with varying behaviors (e.g. activities, transportation modes). Algorithms for integrating information from geographic objects (with the spatial extent of *point, line or regions*) were carefully designed to be generic and accommodate most existing geographic information sources. By virtue of the design, latent context in the stream is exploited to determine *when to apply which algorithm with what sources* – resulting in improving the annotation efficiency of the overall system as well as avoiding information overload. Our experiment demonstrated the effectiveness and efficiency of SeMiTri to act as a semantic platform for diverse trajectory applications. The future research focus is on further enriching the semantic trajectory analytics layer for people trajectories on a large scale.

7. REFERENCES

[1] L. O. Alvares, V. Bogorny, B. Kuijpers, J. Macedo, B. Moelans, and A. Vaisman. A Model for Enriching Trajectories with Semantic Geographical Information. In *GIS*, page 22, 2007.

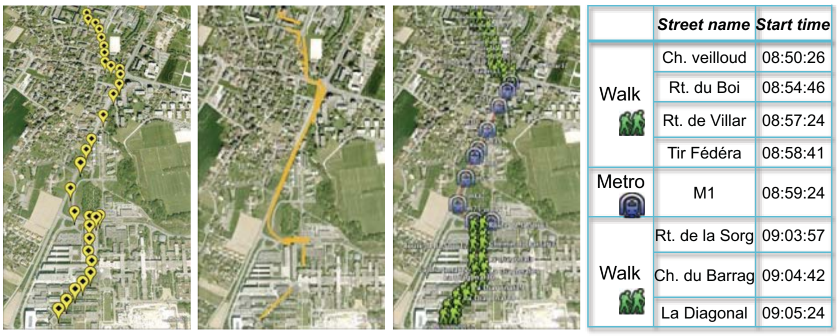
[2] N. Beckmann, H.-P. Kriegel, R. Schneider, and B. Seeger. The R*-Tree: An Efficient and Robust Access Method for Points and Rectangles. *SIGMOD Record*, 19(2):322–331, 1990.

[3] D. Bernstein and A. Kornhauser. *An Introduction to Map Matching for Personal Navigation Assistants.* Princeton University, 1996.

[4] S. Brakatsoulas, D. Pfoser, R. Salas, and C. Wenk. On Map-Matching Vehicle Tracking Data. In *VLDB*, pages 853–864, 2005.

[5] T. Brinkhoff, H.-P. Kriegel, and B. Seeger. Efficient Processing of Spatial Joins using R-Trees. In *SIGMOD*, pages 237–246, 1993.

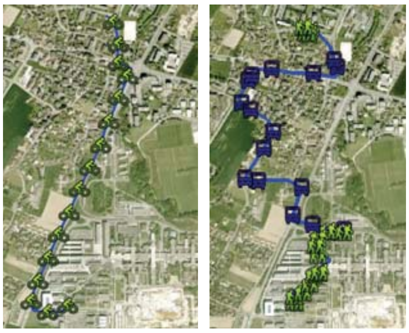
[6] X. Cao, G. Cong, and C. Jensen. Mining Significant



	Street name	Start time
Walk	Ch. veilloud	08:50:26
	Rt. du Boi	08:54:46
	Rt. de Villar	08:57:24
	Tir Fédéra	08:58:41
Metro	M1	08:59:24
Walk	Rt. de la Sorg	09:03:57
	Ch. du Barrag	09:04:42
	La Diagonal	09:05:24

(a) GPS points (b) Map matching (c) Infer transport (d) Move annotation

Figure 15: Move annotation - a home-office example (via Metro)



(a) HomeOffice via Bike (b) HomeOffice via Bus

Figure 16: Home-office (via Bike and via Bus)

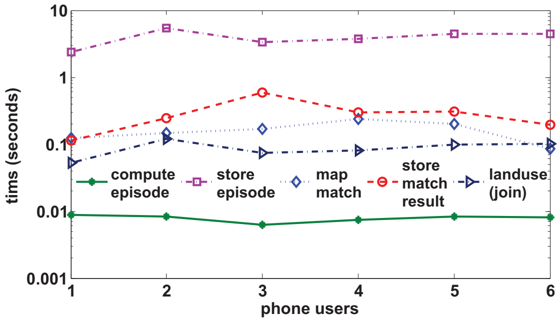


Figure 17: Latency measure

Semantic Locations From GPS Data. In *VLDB*, pages 1009–1020, 2010.

[7] G. D. Forney. The Viterbi Algorithm. *Proceedings of the IEEE*, 61(3):268–278, 1973.

[8] E. Frentzos. *Trajectory Data Management in Moving Object Databases*. PhD thesis, University of Piraeus, 2008.

[9] M. C. González, C. A. Hidalgo, and A. L. Barabási. Understanding Individual Human Mobility Patterns. *Nature*, 453(7196):779–782, 2008.

[10] R. H. Güting and M. Schneider. Realm-Based Spatial Data Types: The ROSE Algebra. *VLDB Journal*, 4:243–286, 1995.

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

[12] J. Han, J.-G. Lee, H. Gonzalez, and X. Li. Mining Massive RFID, Trajectory, and Traffic Data Sets (Tutorial). In *KDD*, 2008.

[13] H. Jeung, M. L. Yiu, X. Zhou, C. S. Jensen, and H. T. Shen. Discovery of Convoys in Trajectory Databases. In *VLDB*, pages 1068–1080, 2008.

[14] N. Kiukkoneny, J. Blom, O. Dousse, D. Gatica-Perez, and J. Laurila. Towards Rich Mobile Phone Datasets: Lausanne Data Collection Campaign. In *ICPS*, 2010.

[15] J. Krumm and E. Horvitz. Predestination: Inferring Destinations from Partial Trajectories. In *Ubicomp*, pages 243–260, 2006.

[16] Z. Li, B. Ding, J. Han, R. Kays, and P. Nye. Mining Periodic Behaviors for Moving Objects. In *KDD*, pages 1099–1108, 2010.

[17] Z. Li, M. Ji, J.-G. Lee, L.-A. Tang, Y. Yu, J. Han, and R. Kays. MoveMine: Mining Moving Object Databases. In *SIGMOD*, pages 1203–1206, 2010.

[18] Y. Lou, C. Zhang, Y. Zheng, X. Xie, W. Wang, and Y. Huang. Map-Matching for Low-Sampling-Rate Gps Trajectories. In *GIS*, pages 352–361, 2009.

[19] M. F. Mokbel and J. J. Levandoski. Toward Context and Preference-Aware Location-Based Services. In *MobiDE*, pages 25–32, 2009.

[20] M. Nergiz, M. Atzori, Y. Saygin, and B. Güç. Towards Trajectory Anonymization: a Generalization-Based Approach. *Transactions on Data Privacy*, 2(1):47–75, 2008.

[21] P. Newson and J. Krumm. Hidden Markov Map Matching Through Noise and Sparseness. In *GIS*, pages 336–343, 2009.

[22] A. T. Palma, V. Bogorny, B. Kuijpers, and L. O. Alvares. A Clustering-based Approach for Discovering Interesting Places in Trajectories. In *SAC*, pages 863–868, 2008.

[23] N. Pelekis, E. Frentzos, N. Giatrakos, and Y. Theodoridis. HERMES: Aggregative LBS via a Trajectory DB Engine. In *SIGMOD*, pages 1255–1258, 2008.

[24] M. A. Quddus, W. Y. Ochieng, and R. B. Noland. Current Map-Matching Algorithms for Transport Applications: State-Of-The Art and Future Research Directions. *Transportation Research Part C*, 15(5):312–328, 2007.

[25] L. R. Rabiner. A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition. *Readings in speech recognition*, pages 267–296, 1990.

[26] S. Spaccapietra, C. Parent, M. L. Damiani, J. A. de Macedo, F. Porto, and C. Vangenot. A Conceptual View on Trajectories. *Data and Knowledge Engineering*, 65:126–146, 2008.

[27] C. E. White, D. Bernstein, and A. L. Kornhauser. Some map matching algorithms for personal navigation assistants. *Transportation Research Part C*, 8(1-6):91–108, 2000.

[28] K. Xie, K. Deng, and X. Zhou. From Trajectories to Activities: a Spatio-Temporal Join Approach. In *LBSN*, pages 25–32, 2009.

[29] Z. Yan, J. Macedo, C. Parent, and S. Spaccapietra. Trajectory Ontologies and Queries. *Transactions in GIS*, 12(s1):75–91, 2008.

[30] Z. Yan, C. Parent, S. Spaccapietra, and D. Chakraborty. A Hybrid Model and Computing Platform for Spatio-Semantic Trajectories. In *ESWC*, pages 60–75, 2010.

[31] Z. Yan, L. Spremic, D. Chakraborty, C. Parent, S. Spaccapietra, and K. Aberer. Automatic Construction and Multi-level Visualization of Semantic Trajectories. In *GIS*, pages 524–525, 2010.

[32] Y. Zheng, Y. Chen, Q. Li, X. Xie, and W.-Y. Ma. Understanding transportation modes based on GPS data for web applications. *Transactions on the Web*, 4(1):1–36, 2010.

[33] Y. Zheng and X. Xie. Learning Location Correlation from GPS Trajectories. In *MDM*, pages 27–32, 2010.

[34] Y. Zheng, L. Zhang, X. Xie, and W.-Y. Ma. Mining Correlation Between Locations Using Human Location History. In *GIS*, pages 472–475, 2009.

[35] C. Zhou, D. Frankowski, P. J. Ludford, S. Shekhar, and L. G. Terveen. Discovering Personally Meaningful Places: an Interactive Clustering Approach. *ACM Transactions on Information Systems*, 25(3):12, 2007.