


A Hybrid Model and Computing Platform for Spatio-semantic Trajectories*

View metadata, citation and similar papers at core.ac.uk

brought to you by  CORE

provided by Infoscience - École polytechnique fédérale de Lausanne

Stefano Spaccapietra¹, and Dipanjan Chakraborty^{2,**}

¹ EPFL, Switzerland
{firstname.surname}@epfl.ch
² IBM Research, India Lab
cdipanjan@in.ibm.com

Abstract. Spatio-temporal data management has progressed significantly towards efficient storage and indexing of mobility data. Typically such mobility data analytics is assumed to follow the model of a stream of (x,y,t) points, usually coming from GPS-enabled mobile devices. With large-scale adoption of GPS-driven systems in several application sectors (shipment tracking to geo-social networks), there is a growing demand from applications to understand the *spatio-semantic* behavior of mobile entities. Spatio-semantic behavior essentially means a semantic (and preferably contextual) abstraction of raw spatio-temporal location feeds. The core contribution of this paper lies in presenting a *Hybrid Model* and a *Computing Platform* for developing a semantic overlay - analyzing and transforming raw mobility data (GPS) to meaningful semantic abstractions, starting from raw feeds to semantic trajectories. Secondly, we analyze large-scale GPS data using our computing platform and present results of extracted spatio-semantic trajectories. This impacts a large class of mobile applications requiring such semantic abstractions over streaming location feeds in real systems today.

1 Introduction

Over the last few years, there has been a tremendous surge in applications and services that exploit real-time location of mobile end points. This is possible in turn due to the large-scale embedding of GPS-driven location sensors in several mobile end points that range from smart phones (e.g. iPhone, Nokia N-series) to specialized GPS chips on shipments, parcels etc. Popular trends suggest large-scale adoption of such mobile end points in future. E.g. Berg Insight (www.berginsight.com) forecasts increase in shipments of GPS-enabled GSM/WCDMA handsets to 770 million units in 2014, representing an coverage rate of 55%. Of course, apart from GPS, other tracking techniques like satellites, radar and RFID also promote location monitoring. Applications and services

* This work is supported by the Swiss FNRS grant 200021-116647/1.

** While the author was in EPFL as Academic Guest.

consuming this spatio-temporal data range from real-time applications to statistical analytics applications. While real-time applications (e.g. location-based advertising) have typically drawn on raw feeds, statistical applications usually require abstracted analytical views on the data for analysis (e.g. geo-marketing, workforce management, study of tele-density by telcos, traffic management etc).

As such, study on raw mobility tracking data has mostly centered around moving object databases and corresponding statistical analytics. Moving object database community has primarily focusing on: (1) Definitions/extensions of trajectory related datatypes such as *moving point* and *moving region* [7][18]; (2) Efficient storage of mobility data, building indexing and querying techniques [3]. Statistical data mining community has on the other hand progressed significantly on approximation functions (e.g. regression or compression) for spatio-temporal mobility data, mining and learning algorithms for pattern discovery [8], primarily considering the raw geometrical perspective on movement.

In this paper, we explore the complimentary challenge that lies in developing appropriate spatio-semantic abstractions of raw spatio-temporal location feeds, catering to a large number of real-time applications. This is because, in recent years, there has been an increase in applications requiring such *abstracted semantic* view on the real-time movement of mobile entities. E.g. Geo-fencing based applications essentially focus on generating high-level events when mobile end points cross domain boundaries or deviate from pre-defined trajectories. Social Networking applications start to exploit real-time location for enriched geo-social collaborations [2] and communications. There is a strong emphasis on developing techniques for higher level, *semantic* events (e.g. Harry just *reached office*, Sally is *shopping in the Owings Mills mall*, Dave is *stuck in traffic* - inferred at varying semantic abstractions from raw GPS-driven location feeds. Solutions used today mostly require human intervention (e.g. applications integrated with twitter) for such semantic (and contextual) abstractions of spatio-temporal data. Our paper is towards providing a model and computing platform for such abstracted spatio-semantic feeds at different levels.

Research relevant to our goal has primarily explored approaches for developing new conceptual models where semantics of movement can be explicitly expressed through application-aware trajectory modeling [16][19][1]. This has resulted in development of high-level semantically meaningful trajectory concepts [5]. However, the primary challenge not yet addressed is to have a generic model to develop these abstracted spatio-semantic trajectories from low-level real-life GPS and other mobility feeds. Apart from handling several issues related to noisy data, a key novelty in our method is to be able to provide a generic set of computing methodologies to represent a spatio-temporal raw data feed at different semantically abstracted levels, starting with basic abstractions (e.g. *stop*, *moves*) to enriched higher-level abstractions (e.g. *office*, *shop*).

To summarize, the core contributions of our paper are: (1) *A hybrid spatio-semantic trajectory model that progressively abstracts high-level semantic concepts from low-level location feeds.* (2) *A computing platform that encapsulates several mobile data abstraction algorithms to enable such semantic enrichment*

of mobility data. (3) Evaluation of the model and computing platform against large-scale real GPS location feeds and presentation of experimental analysis.

2 Related Work

The GeoPKDD (*Geographic Privacy-aware Knowledge Discovery and Delivery*) [5] and MODAP (*Mobility, Data Mining, and Privacy*) projects emphasize the need to develop high-level semantic concepts related to mobility data. However, they focus on data warehousing and mining for trajectories of moving objects, with an aim to preserve privacy of the owner. A systematic approach towards incremental semantic abstraction of raw trajectory data is not addressed.

A body of work exists in defining semantic abstractions over trajectory data [16][1][6][13][19][17]. Spaccapietra et al. [16] provide a trajectory structure as a sequence of *moves* and *stops* in between, with *begin* and *end* events to represent a trajectory. Work also exists on analysis of tourist movements [1] and semantic interpretation of stops [6] and moves [13]. In addition, reasoning has been applied to mobility data: Yan et al. design ontologies for conjunctive query processing over trajectory-related knowledge [19]; Wessel et al. [17] propose a situational reasoning engine to recognize events from trajectories with description logic.

While these papers focus on definitions and interpretation/analysis of semantic trajectory models relevant for different application domains, the ability to infer several such basic semantic concepts at different abstracted levels from real GPS feeds has remained a challenge. Our model and computing platform enables this. We carefully adopt key concepts from related literature in semantic trajectory modeling for our spatio-semantic representation of trajectories. Further, we take a computational perspective on the spatio-temporal data for extraction of such key concepts and define a conceptual model (called *episodes*) that encapsulates several such semantic trajectory concepts (e.g. *stop*, *move*, *begin*, *end*). This enables us to process raw GPS events in our computing platform.

Complementary to semantic trajectory representation literature, another body of related work on trajectory data analysis is in applying conventional data mining and machine learning methods for clustering [12], classification [11], outlier detection [10], finding convoys [9] and sequential rule-driven pattern mining [4], over real GPS data feeds. The common goal of such statistical analytics is to extract knowledge about trajectories in terms of patterns. Interestingly, the patterns extracted from this body are represented on raw mobility data and are disconnected from the associated *semantic* interpretation of such data.

To the best of our knowledge, our work is the first to bridge the gap between these two bodies of research and presents a model and computing platform for spatio-semantic knowledge discovery from GPS feeds.

3 Spatio-semantic Trajectory Model

As discussed, related work has focused either on high-level semantic representation models of user's mobility or low-level analytics on spatio-temporal GPS

data. Our proposal is a hybrid *Spatio-Semantic* trajectory model that: (1) encapsulates raw GPS spatio-temporal trajectory data; (2) allows for progressive abstraction of the raw data to higher-level semantic representation of such data; (3) encapsulates well-known concepts used in literature for trajectories, e.g. stop-move in [16]. The model is usable by several applications requiring varying degrees of such comprehensive trajectory representations from data as well as semantic perspectives (and hence hybrid). Our key design considerations are:

- *Raw Data characteristics*: Model should consider characteristics of raw mobility tracking data (e.g. spatial and temporal gaps, uncertainties) to create simple *low-level* representations (e.g. hourly, daily, monthly and geo-fenced trajectories)
- *Progressive computation possible*: Model should be designed so that a layered computing platform can systematically generate higher-level semantic abstractions from underlying lower-level trajectory representations

Therefore, our hybrid model consists of (1) *Data Model*: to encapsulate the trajectory definitions available from raw data perspective; (2) *Conceptual Model*: a key mid-level abstraction of a trajectory that allows for progressive abstraction of the raw mobility data; (3) *Semantic Model*: to encapsulate spatio-semantic behavior of the trajectory. Fig. 1 provides a high-level view of such models.

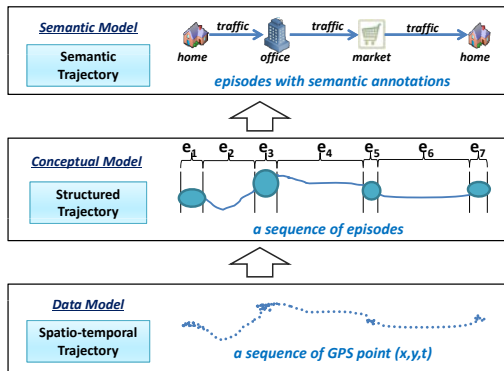


Fig. 1. The Hybrid Spatio-Semantic Trajectory Model

3.1 Data Model

Data model is the first abstraction level over the raw mobility (GPS) data. The raw mobility data is traditionally captured by mobile location sensors, and continuously records the evolving position where a moving object temporally resides. So, raw mobility data is essentially as a long sequence of spatio-temporal tuples (*position, timestamp*) collected over varying time lengths. Most real-life location traces today are essentially GPS-like tuples (*longitude, latitude, timestamp*), (x, y, t) in short. From now on, we use the term *GPS feed* to represent the *raw* sequence of (x, y, t) spatio-temporal mobility data.

In our Data Model, we decompose each GPS feed into *subsequences* so that each such subsequence represents one meaningful unit of movement. We call these meaningful units “*spatio-temporal trajectories*”. Consequently, a spatio-temporal trajectory has a starting point (x, y, t) and an ending point (temporally ordered) that delimits the subsequence, along with a time interval $[t_{begin}, t_{end}]$.

Definition 1 (Spatio-temporal Trajectory \mathcal{T}_{spa}). *A spatio-temporal trajectory \mathcal{T}_{spa} is a cleansed subsequence of raw GPS feed for a given moving object in a given time interval $[t_{begin}, t_{end}]$. It is a list of triples (x, y, t) which is ordered by increasing t , i.e. $\mathcal{T}_{spa} = \{p_1, p_2, \dots, p_n\}$ where $p_i = (x_i, y_i, t_i)$ represents a spatio-temporal point.*

Several issues are relevant for systematic computation of a sequence of \mathcal{T}_{spa} from a computational perspective. E.g. the raw GPS feed need to be cleansed, missing points interpolated and errors in data acquisition corrected. We provide such *Data Preprocessing* methods in our computing platform in Section 4.1. A key concept in computing this model is to identify the meaningful *dividing points* in a raw GPS feed that separates two temporally ordered \mathcal{T}_{spa} . A dividing point identifies the *end* of one trajectory \mathcal{T}_{spa} and the *begin* of the next one. Note that the exact begin and end coordinates may not be co-located (e.g. due to data collection gaps). Typical examples of such dividing points could be temporal (daily, hourly trajectories) or spatial (trajectory of a car in a city) or places in the raw feed where there are large spatial or temporal disconnections. Section 4.2 describes the policy for identifying such dividing points and computing \mathcal{T}_{spa} .

3.2 Conceptual Model

Intuitively, conceptual model is the logical partitioning of a *single* spatio-temporal trajectory \mathcal{T}_{spa} into a series of non-overlapping temporally separated *episodes*. A \mathcal{T}_{spa} having such annotated episodes is called a *Structured Trajectory* (\mathcal{T}_{str}).

An *episode* abstracts those sequences of spatio-temporal tuples that show a high degree of correlation w.r.t. some identifiable feature (e.g. velocity, angle of movement, density, time interval etc). This generic *structural* representation enables us to compute such sequences using structured techniques (described in Section 4.3). An *episode* has the following salient features:

- *Encapsulates semantic trajectory concepts:* High-level trajectory concepts such as *begin, end, stops, moves* [16] become *sub-classes* of *episode*. Moreover, it also encapsulates additional meaningful trajectory concepts such as *jumps, pattern-driven movement* sequences (not necessarily *stop* or *move*), which is defined in literature for domains such as trajectory of wild life [14].
- *Computed automatically:* Episodes can be computed with relevant *Trajectory Structure* algorithms. This is because the correlations are essentially geometric characteristics of GPS feed like *velocity, acceleration, orientation, density*, or other *spatio-temporal* correlations.
- *Enables Data Compression:* Instead of semantic annotation of each GPS records directly (which is possible), episodes essentially enable single semantic tagging of correlated GPS tuples having similar features. This reduces the data size to represent trajectories at the conceptual level. E.g. Fig. 1 shows semantic annotation of seven episodes in the conceptual model which is more efficient than annotation of each GPS tuple in the data model.

Definition 2 (Structured Trajectory \mathcal{T}_{str}). A structured trajectory \mathcal{T}_{str} consists of a sequence of trajectory units, called “episode”, i.e. $\mathcal{T}_{str} = \{e_1, e_2, \dots, e_m\}$

- A episode (e) groups a subsequence of \mathcal{T}_{spa} with k consecutive GPS points having some similar characteristics $\{p_1^{(e_i)}, \dots, p_k^{(e_i)}\}$ derived from \mathcal{T}_{spa} .
- For data compression, episode is a tuple that stores only the subsequence’s temporal duration and spatial extent. $e_i = (time_{from}, time_{to}, bounding_{rectangle}, center)$.

3.3 Semantic Model

In *Semantic Model*, a trajectory \mathcal{T}_{sem} is an annotated enhancement of a *structured trajectory* \mathcal{T}_{str} enriched with semantic knowledge. Such annotations can be made on episodes in the \mathcal{T}_{str} as well as on the whole \mathcal{T}_{str} .

A typical example of a *Semantic Model* is in Fig. 1 (upper layer), where GPS feed are enriched with an employee’s spatio-semantic movement pattern: he goes to work from *home* (morning); after *work* (later afternoon), he leaves for shopping in *market*, and finally reaches *home* (evening).

Our model is designed so that it can integrate data from third party sources (e.g. landuse data, road network, points of interest), or social network data related to locations. The model instances can even be inferred through learning patterns from underlying GPS feeds. Our computing platform describes semantic enrichment methodologies that can be applied on such third party data towards computing instances of the semantic model (Section 4.4).

Definition 3 (Semantic Trajectory \mathcal{T}_{sem}). A semantic trajectory \mathcal{T}_{sem} is a structured trajectory with added semantic annotation: episodes are enriched in terms of semantic episodes (se) with geographic or application knowledge. i.e.

$\mathcal{T}_{sem} = \{se_1, se_2, \dots, se_m\}$, where semantic episode $se_i = (sp_i, t_{in}^{(sp_i)}, t_{out}^{(sp_i)})$

- sp_i (semantic position) is a meaningful location object, which can be real-world objects from geographic knowledge (e.g. building, roadSegment, administrativeRegion, landuse), or more application domain knowledge (e.g. home, office that belong to specific people in a given application).
- $t_{in}^{(sp_i)}$ is the incoming timestamp for trajectory entering this semantic position (sp_i), and $t_{out}^{(sp_i)}$ is the outgoing timestamp for trajectory leaving sp_i . They can be approximated by $time_{from}$ and $time_{to}$ in episode.
- From data compression point of view, many episodes in one or more structured trajectories \mathcal{T}_{str} can be located in the same sp_i in \mathcal{T}_{sem} .

Our hybrid model is generic and different ontological frameworks for trajectory modeling [19] [17] can be represented with such model. We do not describe additional examples, but focus on the second contribution - a computing platform that enables population of our hybrid model instances from live GPS feeds.

4 Trajectory Computing Platform

The *Trajectory Computing Platform* exploits the Spatio-Semantic Trajectory model and build trajectory instances of the models at every level (*spatio-temporal*,

structural, semantic), from large-scale real-life GPS feed. Fig. 2 shows the four layers in our platform, each containing several techniques for progressive computation of the trajectory instances.

- 1) **Data Preprocessing Layer:** This layer cleanses the raw GPS feed, in terms of preliminary tasks such as outliers removal and regression-based smoothing. The outcome of this step is a cleansed sequence of (x, y, t) .
- 2) **Trajectory Identification Layer:** This layer divides the cleansed (x, y, t) GPS spatio-temporal points into several meaningful subsequences (spatio-temporal trajectories T_{spa}). This step exploits gaps present in the underlying data and in addition, exploits well-defined polices for temporal and spatial demarcations (e.g. time interval for a day, week, city area etc).
- 3) **Trajectory Structure Layer:** This layer is responsible for computing *episodes* present in each spatio-temporal trajectory and generates structured trajectories T_{str} . It contains several algorithms for computing correlations between temporally occurring sequence of GPS points.
- 4) **Semantic Enrichment Layer:** This layer semantically annotates T_{str} and computes semantic trajectories T_{sem} . It integrates episodes with relevant semantic data from 3rd party sources, and includes algorithms for such integration. E.g. we use spatial join for inferring semantic regions (e.g. landuses), map-matching for inferring road networks, and hidden Markov model for inferring semantic points (e.g. points of interests).

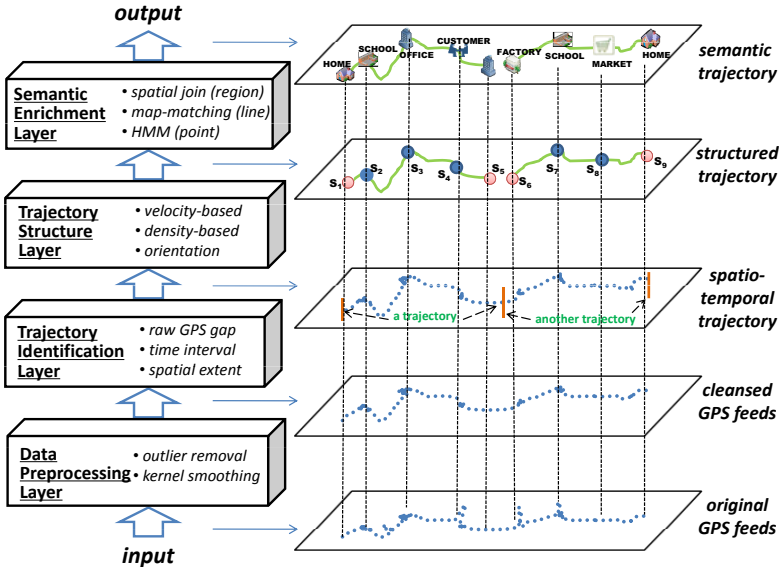


Fig. 2. Spatio-Semantic Trajectory Computing Platform

4.1 Data Preprocessing Layer

Due to GPS measurements and sampling errors from mobile devices, the recorded position of a moving object is not always precise or correct [20]. There is work on determining possible causes for such uncertainty [3]. Data of spatio-temporal tracks from mobile sensors in real world is usually unreliable, imprecise, incorrect and contain noisy records. The *Data Preprocessing Layer* has techniques to clean this data before fitting our *hybrid* model on the data.

Adopting GPS preprocessing techniques for data cleaning [15], we have built techniques to detect (1) *outliers*: observations that deviate significantly from the desired correct position; (2) *random noise*: GPS signals can have noise from several sources. E.g. ionospheric effects and clocks of satellites can contribute towards white noise of ± 15 meters (www.kowoma.de/en/gps/errors.htm).

For *outliers*, we adopt threshold-driven techniques on velocity and remove points that do not give us a reasonable correlation with expected velocity. Each GPS feed has domain knowledge of the moving object (e.g. car, human, cycle etc). This lets us compute such speed thresholds from the data. For *random noises*, we design a Gaussian kernel based local regression model to smooth out the GPS feed. The smoothed position (\hat{x}, \hat{y}) is the weighted local regression based on the past points and future points within a sliding time window, where the weight is a Gaussian kernel function $k(t_i)$ with the kernel bandwidth σ (Formula 1). To control the smoothing related information loss, we adopt a reasonably small value for σ (e.g. $5 \times$ GPS sampling frequency) so that only nearby points can affect the smoothed position. This is necessary as we wanted to calibrate the technique to handle only the noise while avoiding under-fitting.

$$(\hat{x}, \hat{y}) = \frac{\sum_i k(t_i)(x_{t_i}, y_{t_i})}{\sum_i k(t_i)}, \text{ where } k(t_i) = e^{-\frac{(t_i-t)^2}{2\sigma^2}} \quad (1)$$

Fig. 3, 4, 5 show an example of our smoothing algorithm on a real data set taken from wild-life tracking data on 15.03.2008. It contains 52 GPS (x,y,t) records. Fig. 3 shows the smoothed longitude (X in cartesian coordinate). Fig. 4 shows the smoothed latitude (Y in cartesian coordinate) and Fig. 5 plots the original GPS feed before smoothing and the smoothed one.

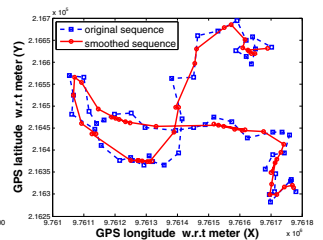
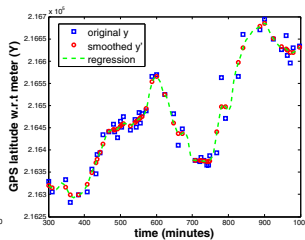
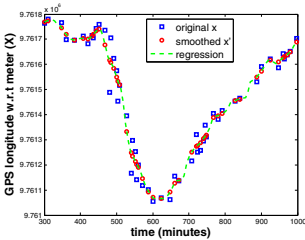


Fig. 3. Smooth GPS (x)

Fig. 4. Smooth GPS (y)

Fig. 5. Original/smoothed

4.2 Trajectory Identification Layer

This layer uses the processed data and extracts relevant non-overlapping spatio-temporal trajectories \mathcal{T}_{spa} (*data model*). The central issue here is to determine reasonable identification policies, to identify *dividing points* (x_i, y_i, t_i) (introduced earlier) and divide the continuous GPS sequence into consecutive trajectories at appropriate positions. We present three types of trajectory identification policies we have implemented for various trajectory scenarios: *Raw GPS Gap*, *Predefined Time Interval* and *Predefined Spatial Extent*.

Policy 1 (Raw GPS Gap). *Divide the (x, y, t) sequence into several spatio-temporal trajectories according to the raw GPS gaps.*

- (1) *Given a large time interval $\Delta_{duration-large}$, for any two consecutive GPS records $p_i(x_i, y_i, t_i)$ and $p_{i+1}(x_{i+1}, y_{i+1}, t_{i+1})$, if the temporal gap $t_{i+1} - t_i > \Delta_{duration-large}$, then p_i is the ending point of current trajectory whilst p_{i+1} is the starting point of the forthcoming trajectory.*
- (2) *Given both time interval $\Delta_{duration}$ and spatial distance $\Delta_{distance}$, for any two consecutive GPS records p_i and p_{i+1} , if the temporal gap $t_{i+1} - t_i > \Delta_{duration}$ and the spatial gap $\sqrt{(x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2} > \Delta_{distance}$, then p_i is the ending point of current trajectory whilst p_{i+1} is the starting point of the forthcoming trajectory.*

This policy used significant temporal (and spatial) gaps in GPS feed to logically separate two consecutive spatio-temporal trajectories. This is because GPS-alike tracking devices are usually turned off (automatically or manually) when the object does not move for a long while (e.g. to save power). The first sub-policy exploits large temporal gaps $\Delta_{duration-large}$ to extract spatio-temporal trajectories. This is typically relevant for vehicle movement scenarios. E.g. our dataset of 17,241 car GPS traces (2,075,213 GPS records) resulted in 83,134 spatio-temporal trajectories. The second sub-policy uses both temporal and spatial gaps, where the two parameters are determined by statistical analysis of GPS feeds (e.g. gap distribution, type of movement - vehicular, pedestrian etc).

Policy 2 (Predefined Time Interval). *Divide the stream of GPS feed into several subsequences contained in given time intervals, e.g. hourly trajectory, daily trajectory, weekly trajectory, monthly trajectory.*

This policy allows us to meaningfully divide a GPS feed into periods for analyzing mobility behaviors. Short-term period is particularly relevant for human movements. Wild-life monitoring on the other hand usually requires longer-term trajectory behaviors such as monthly or seasonal patterns.

Policy 3 (Predefined Space Extent). *Divide the stream of GPS feed into several subsequences according to a spatial criteria, e.g. fixed distance, geo-fenced regions, movement between predefined points in network constrained trajectories.*

This policy allows us to divide a GPS feed in terms of fixed spatial extent; in a specific domain zone (e.g. Lausanne downtown), where trajectories ought to be created according to their entering and leaving the zone; or between two given positions of crossings.

4.3 Trajectory Structure Layer

After identifying spatio-temporal trajectories, the next task is to compute their internal structures, and to construct structured trajectories \mathcal{T}_{str} consisting of meaningful episodes. The core issue in *trajectory structure* is to group continuous GPS points into an episode. We have implemented *velocity*, *density*, *orientation* and *time series* based algorithms for identifying episodes. In this paper, the focus is on the whole trajectory data computing platform. Thus we only address a representative one, i.e. *velocity-based* approach, which we have found quite useful in analyzing our current data sets.

In this approach, we focus on two kinds of episodes (i.e. *stops* and *moves*). The idea is to determine whether a GPS point $p(x, y, t)$ belongs to a stop episode or a move episode by using a speed threshold (Δ_{speed}). Hence, *if the instant speed of p is lower than Δ_{speed} , it is a part of a stop, otherwise it belongs to a move*. Fig. 6 traces the speed evolution of a vehicle, showing how stops can be determined by a given Δ_{speed} . Besides Δ_{speed} , we also use a second parameter - *minimal stop time* τ in order to avoid false positives (e.g. short-term *congestions* with low velocity should not be a stop episode).

Determining the value for Δ_{speed} is a challenge: *if Δ_{speed} is too high, many stops appear; on the contrary, if Δ_{speed} is too low, probably no stops are computed*. Fig. 6 simply shows a constant Δ_{speed} applied all across the trajectory. This is not practical in real-world scenarios, where Δ_{speed} should rather be dynamic according to the context of the moving object. For example, vehicles with different levels of performance (bicycles or motor cars), different road networks (a highway or a secondary road), different weather conditions (sunny or snowy days) call for diverse speed thresholds. However, it is not easy to get these contexts. To avoid this, we design a generic method for determining Δ_{speed} , based on class of moving objects being monitored (which is available) and the real-time underlying movement area. This can be applied to most real-life GPS feeds.

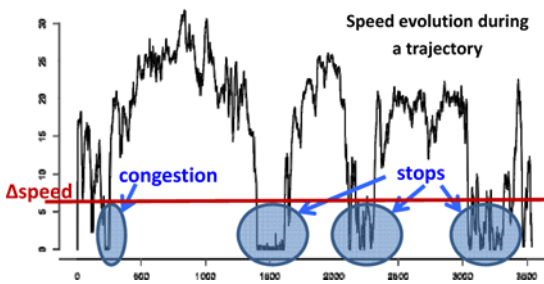


Fig. 6. Velocity-based Stop Identification

Definition 4 (Dynamic Velocity Threshold - Δ_{speed}). For each GPS point $p(x, y, t)$ of a given moving object (obj_{id}), the Δ_{speed} is dynamically related to the moving object (by $objectAvgSpeed$ - the avg. speed of this moving object) and the $positionAvgSpeed$ - the

avg. speed of moving objects in this position $\langle x, y \rangle$; i.e. $\Delta_{speed} = \min\{\delta_1 \times \overline{objectAvgSpeed}, \delta_2 \times \overline{positionAvgSpeed}\}$

$\overline{objectAvgSpeed}$ is easy to calculate. We approximate $\overline{positionAvgSpeed}$ through space division. We divide the mobility space into regular cells (or directly using the available landuse data) and calculate the average speed in each cell $\overline{cellAvgSpeed}$. Algorithm 1 provides the pseudocode to determine Δ_{speed} . We analyze sensitivity of the coefficients δ (e.g. 30%) through experiments.

Algorithm 1. `getDynamic Δ_{speed}` (`gpsPoint`, `obj id` , δ)

input : `gpsPoint` $p = (x, y, t)$, moving object `obj id`
output: dynamic speed threshold Δ_{speed}
1 get the average speed of this moving object `obj id` : $\overline{objectAvgSpeed}$;
2 get the average speed of the cell that (x,y) belongs to: $\overline{cellAvgSpeed}$;
3 compute the dynamic speed threshold by Definition 4;
4 return Δ_{speed}

In some scenarios, GPS tracking data have instant speeds (s) values captured by the devices. We use them for calculating Δ_{speed} and identifying stops; otherwise, s is approximated by the average speed between the previous spatio-temporal point $(x_{i-1}, y_{i-1}, t_{i-1})$ and the next one $(x_{i+1}, y_{i+1}, t_{i+1})$, i.e. $s_i = \frac{\|(x_{i+1}, y_{i+1}) - (x_{i-1}, y_{i-1})\|_2}{t_{i+1} - t_{i-1}}$. This is possible as GPS data is usually sampled frequently (e.g. a few minutes or even every second).

Algorithm 2 provides the pseudocode for determining *velocity-based trajectory structure*: firstly, we compute the instant speed if not available from GPS devices; secondly, we compute the dynamic Δ_{speed} (using Algorithm 1) and annotate the GPS point with ‘M’ or ‘S’ tag; finally, stops and moves are computed with the consecutive same tags, using preconditions on minimal stop time τ .

4.4 Semantic Enrichment Layer

Once structured trajectories \mathcal{T}_{str} are computed in terms of episodes such as *stop* and *move*, the final task is to enrich their semantics by integrating these spatio-temporal episodes with semantic knowledge, and create the model for spatio-semantic trajectories \mathcal{T}_{sem} . This is enabled through the Semantic Enrichment Layer. This layer utilizes available third party data sources to gather additional context about each episode.

We have designed three typical algorithms to be able to integrate landuse data, road networks and maps as well as information about points of interest with each episode in the structured trajectory. Landuse data is integrated using *spatial join* with semantic regions for computing $\mathcal{T}_{sem}^{(region)}$; *Map-matching* algorithm is used with semantic lines (road networks) for computing $\mathcal{T}_{sem}^{(line)}$ and *hidden Markov model (HMM)* is used with semantic points (e.g. points of interests) for computing $\mathcal{T}_{sem}^{(point)}$. In this paper, we only illustrate the first case: $\mathcal{T}_{sem}^{(region)}$.

Algorithm 2. Velocity-based trajectory structure

```

Input: a raw trajectory  $\mathcal{T}_{raw} = \{p_1, p_2, \dots, p_n\}$ 
Output: a structured trajectory  $\mathcal{T}_{str} = \{e_1, e_2, \dots, e_m\}$  where  $e_i$  is a tagged trajectory episode (stop  $S$  or move  $\mathcal{M}$ )

1 begin
2   /* initialize: calculate GPS instant speed if needed */
3   ArrayList( $x, y, t, tag$ )  $gpsList \leftarrow getGPSList(\mathcal{T}_{spa})$ ;
4   if no instant speed from GPS device then
5     | compute GPS instant speed  $s_i$  for all  $p_i = (x, y, t) \in gpsList$ ;
6   /* episode annotation: tag each GPS point with 'S' or 'M' */
7   forall  $p_i = (x, y, t) \in gpsList$  do
8     | // get dynamic  $\Delta_{speed}^{(i)}$  by Algorithm 1
9     |  $\Delta_{speed}^{(i)} \leftarrow getDynamic\Delta_{speed}(p, objid, \delta)$ ;
10    | // tag GPS point as a stop point 'S' or a move point 'M'
11    | if instant speed  $s_i < \Delta_{speed}^{(i)}$  then
12    |   | tag current point  $p_i(x, y, t)$  as a stop point 'S';
13    |   else
14    |     | tag current point  $p_i(x, y, t)$  as a move point 'M';
15  /* compute episodes: grouping consecutive same tags */
16  forall consecutive points with the same tag 'S' do
17    | // compute stop episode
18    | get the total time duration  $t_{interval}$  of these points;
19    | if  $t_{interval} > \tau$  the minimal possible stop time then
20    |   |  $stop \leftarrow (time_{from}, time_{to}, center, boundingRectangle)$ ;
21    |   |  $\mathcal{T}_{str}.(stop, 'S')$ ; // add the stop episode
22    |   else
23    |     | change the 'S' tag to 'M' for all these points; // as "congestion"
24  forall consecutive points with the same tag 'M' do
25    | // compute move episode
26    |  $move \leftarrow (stop_{from}, stop_{to}, duration)$  // create a move episode
27    |  $\mathcal{T}_{str}.(move, 'M')$ ; // add the move episode
28  return the structured trajectory  $\mathcal{T}_{str}$ ;
29 end

```

We use *spatial join* to compute $\mathcal{T}_{sem}^{(region)}$. *Spatial join* is used to calculate the topological correlations between episodes and semantic regions. For each episode (e.g. stop), we use either the spatial *bounding rectangle* of the episode or the *center* to do spatial join. Further, we apply landuse data that describes the use of natural environment to tag each episode. To do this, we divide the landuse space into cells (depending on density) and correlate each episode to available landuse

T1: Settlement and urban areas

- Industrial and commercial area
- Building area
 - Residential, public building ...
- Transportation areas
 - Road, railway, airport
- Special urban areas
- Recreational and cemeteries
 - Public parks, sports, camping ...

T2: Agricultural areas

- Orchard, vineyard, arable land ...

T3: Wooded areas

- Forest, brush forest, woods ...

T4: Unproductive areas

- Lakes, rivers, ...

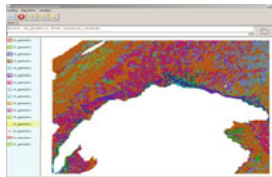


Fig. 7. Landuse Ontology

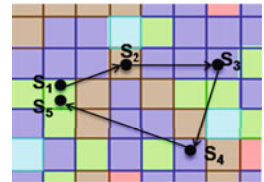


Fig. 8. Real Landuse

Fig. 9. Synthetic Landuse

cells. As an example, Fig. 8 shows the landuse about Lausanne downtown, where we divided the space into $100 \times 100 m^2$ cells. Each cell has semantic annotations from the landuse ontology in Fig. 7 from Swisstopo (www.swisstopo.admin.ch).

For some trajectory scenarios, when real landuse cells are not available, we generate Gaussian distributed synthetic cells. Fig. 9 shows a sample trajectory with five stop episodes. S_1 and S_5 share the same landuse; S_3 and S_4 belong to the same landuse category (shown in same color). In this case, semantic trajectory \mathcal{T}_{sem} is a sequence of four landuse cells with a temporal duration ($cell_{id}, time_{in}, time_{out}$).

5 Experiment Analysis

We have validated our model and computing platform against different kinds of real-life GPS feed. Table 1 provides a short summary of the data used in our experimental analysis. Milan car and Laussane taxi are two big datasets from the GeoPKDD project and Swisscom (www.swisscom.ch) respectively. In addition, we use two public Athens datasets from R-tree portal (www.rtreeportal.org).

Table 1. Trajectory datasets - real life GPS feed

	<i>Dataset</i>	<i># objects</i>	<i># GPS records</i>	<i>Traking time</i>	<i>Sampling frequency</i>
(1)	car (Milan)	17,241	2,075,213	1 week	avg. 40 seconds
(2)	bus (Athens)	2	66,095	108 days	30 seconds
(3)	truck (Athens)	50	112,203	33 days	30 seconds
(4)	taxi (Lausanne)	2	3,347,036	5 months	1 second

In order to present trajectory computing results, we have implemented a hybrid trajectory visualization tool using Java 2D API. Fig. 10 provides a snapshot of the tool presenting four sub-figures corresponding to original *GPS feeds*, *spatio-temporal trajectory*, *structured trajectory*, and *semantic trajectory* computed from the truck dataset. The order of sub-figures (from left to right) follows the progressive computation of trajectories from the raw feed.

- Sub-figure (a) visualizes the spatial locations of 112,203 raw GPS records, in terms of their 2D geometric coordinates (x, y) , without any further meanings (output of *Data Preprocessing Layer*).
- Sub-figure (b) shows 310 spatio-temporal trajectories obtained from the (x, y, t) cleansed sequence (output of *Trajectory Identification Layer*). To distinguish, the neighboring trajectories are shown in different colors.
- Sub-figure (c) displays the trajectory episodes (i.e. stops and moves) and visualizes structured trajectories (output of *Trajectory Structure Layer*). There are 1826 stops (visualized as *points*) and 1849 moves (as *lines between points*).
- Sub-figure (d) shows the output of *Semantic Enrichment Layer*. It displays enriched stop episodes, where 1826 stops are mapped to 160 landuse cells (visualized as *squares*) in 5 types. Cells from different types are drawn with distinct colors.

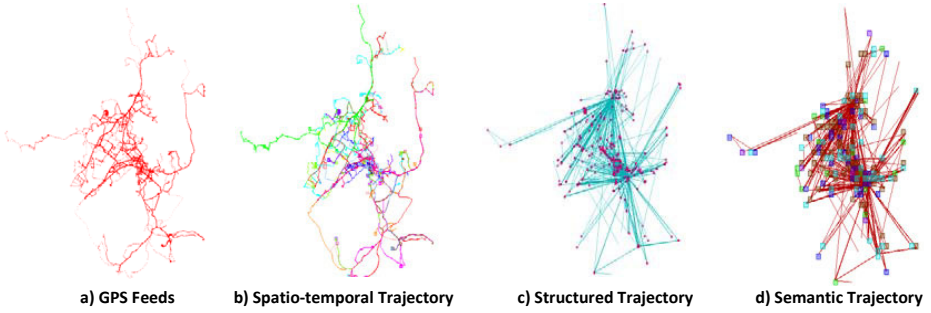


Fig. 10. Visualization - from GPS feed to semantic trajectories

An interesting aspect to observe is the decrease in the data size as trajectories are abstracted to the higher level model. To demonstrate this, we compute the *semantic abstraction rate* as $\log_2\left(\frac{\#GPS}{\#dataComputed}\right)$, where $\#GPS$ is the number of the initial GPS records and $\#dataComputed$ is the number of computed model instances, i.e. the number of trajectories, episodes (stops and moves), and semantic episodes (e.g. landuse cells). For example we observe that for taxi dataset, 3,347,036 GPS records are abstracted to 1,145 trajectories with 1,874 stops and 2,925 moves in structured trajectories, and even lesser 816 semantic stops in semantic trajectories. This is because the higher layer trajectory encapsulates multiple concepts from the underlying lower layer trajectory. Fig. 11 shows the abstraction results for the four datasets.

Another interesting (and reasonable) observation is that the abstraction rate is proportional to the GPS sampling frequency. From left to right in Fig. 11, the GPS recording frequency is respectively one record per 40 seconds (on average), 30 seconds, 30 seconds, and one second. We see the higher recording frequency (like taxi data), the more is the compression (i.e. the higher abstraction rate).

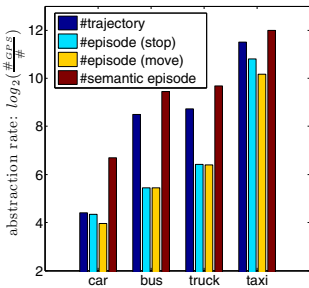


Fig. 11. Different levels of data abstraction

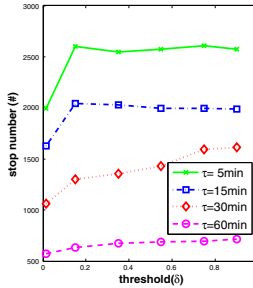


Fig. 12. Δ_{speed} w.r.t. total stop number

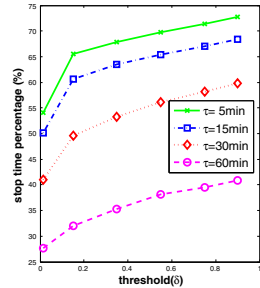


Fig. 13. Δ_{speed} w.r.t. total stop time

As mentioned before, the coefficients for speed thresholds play a role in determining the number of stop and move episodes and as pointed out before, is

dependent on several factors (vehicle type, road type etc). Results presented in Fig. 11 have used the same coefficient of speed thresholds ($\delta_1 = \delta_2 = \delta = 0.3$) and the same minimal stop duration ($\tau = 15 \text{ mins}$) to provide a comparative picture of the abstraction. However, these parameters effect the number of trajectory episodes and needs to be calibrated accordingly.

We analyzed the sensitivity of δ and τ in identifying stop episodes. Fig. 12 shows the number of stops we get with different δ and τ for the Athens truck data. With higher τ (from five minutes to one hour), the number of stops decreases from 2601 to about 633 when given $\delta = 0.15$; whilst with higher δ (from 0.015 to 0.9), the stop number goes up then saturates, because stops computed with higher coefficient δ (i.e. higher Δ_{speed}) usually have longer duration. Therefore stop number decrease as some stops join together. Nevertheless, we observe that the total percentage of time duration for stops always increases when the minimal stop time τ becomes smaller or the speed threshold δ increases (see Fig. 13). We are investigating means to dynamically calibrate these parameters.

6 Conclusion

In this paper, we propose a hybrid spatio-semantic model and a computing platform for trajectories of moving objects. Our hybrid model can represent trajectories in terms of both spatio and semantic mobility characteristics, supporting different levels of data abstraction. Through experimental analysis of real-life GPS feeds, we demonstrate how our model and platform achieve the purpose of progressive abstraction of the raw mobility data. We present spatio-semantic trajectory computing results in various live mobility feeds, and present insights on parameters that guide the sensitivity of such computing platform. This approach can be applied to other location feeds like cellular location data. Our future work focuses on inferring spatio-semantic trajectories from diverse location (and sensory) sources.

References

1. Alvares, L.O., Bogorny, V., Kuijpers, B., Macedo, J., Moelans, B., Vaisman, A.: A Model for Enriching Trajectories with Semantic Geographical Information. In: ACM-GIS, p. 22 (2007)
2. Banerjee, N., Chakraborty, D., Dasgupta, K., Mittal, S., Nagar, S., Saguna: R-U-In? - Exploiting Rich Presence and Converged Communications for Next-Generation Activity-Oriented Social Networking. In: MDM, pp. 222–231 (2009)
3. Frentzos, E.: Trajectory Data Management in Moving Object Databases. PhD thesis, University of Piraeus (2008)
4. Giannotti, F., Nanni, M., Pinelli, F., Pedreschi, D.: Trajectory Pattern Mining. In: KDD, pp. 330–339 (2007)
5. Giannotti, F., Pedreschi, D.: Mobility, Data Mining and Privacy - Geographic Knowledge Discovery. Springer, Heidelberg (2008)
6. Gómez, L., Vaisman, A.: Efficient Constraint Evaluation in Categorical Sequential Pattern Mining for Trajectory Databases. In: EDBT, pp. 541–552 (2009)

7. Güting, R., Schneider, M.: *Moving Objects Databases*. Morgan Kaufmann, San Francisco (2005)
8. Han, J., Lee, J.-G., Gonzalez, H., Li, X.: Mining Massive RFID, Trajectory, and Traffic Data Sets. In: *KDD Tutorial* (2008)
9. Jeung, H., Yiu, M.L., Zhou, X., Jensen, C.S., Shen, H.T.: Discovery of Convoys in Trajectory Databases. In: *VLDB*, pp. 1068–1080 (2008)
10. Lee, J.-G., Han, J., Li, X.: Trajectory Outlier Detection: A Partition-and-Detect Framework. In: *ICDE*, pp. 140–149 (2008)
11. Lee, J.-G., Han, J., Li, X., Gonzalez, H.: TraClass: Trajectory Classification Using Hierarchical Region-Based and Trajectory-Based Clustering. In: *VLDB*, pp. 1081–1094 (2008)
12. Lee, J.-G., Han, J., Whang, K.-Y.: Trajectory Clustering: a Partition-and-Group Framework. In: *SIGMOD*, pp. 593–604 (2007)
13. Mouza, C., Rigaux, P.: Mobility Patterns. *GeoInformatica* 9(4), 297–319 (2005)
14. Santer, R.D., Yamawaki, Y., Rind, F.C., Simmons, P.J.: Motor Activity and Trajectory Control During Escape Jumping in the Locust *Locusta Migratoria*. *Journal of Comparative Physiology A* 191(10), 965–975 (2005)
15. Schüssler, N., Axhausen, K.: Processing GPS Raw Data Without Additional Information. *Transportation Research* 8 (2009)
16. Spaccapietra, S., Parent, C., Damiani, M.L., de Macedo, J.A., Porto, F., Vangenot, C.: A Conceptual View on Trajectories. *Data and Knowledge Engineering* 65, 126–146 (2008)
17. Wessel, M., Luther, M., Möller, R.: What Happened to Bob? Semantic Data Mining of Context Histories. In: *Description Logics* (2009)
18. Wolfson, O., Xu, B., Chamberlain, S., Jiang, L.: Moving Objects Databases: Issues and Solutions. In: *SSDBM*, pp. 111–122 (1998)
19. Yan, Z., Macedo, J., Parent, C., Spaccapietra, S.: Trajectory Ontologies and Queries. *Transactions in GIS* 12(1), 75–91 (2008)
20. Zhang, J., Goodchild, M.F.: *Uncertainty in Geographical Information*, 1st edn. CRC, Boca Raton (2002)