Collecte, intégration et visualisation de données Web incertaines sur des objets mobiles

Mouhamadou Lamine Ba Sébastien Montenez
Talel Abdessalem Pierre Senellart

Institut Mines–Télécom ; Télécom ParisTech ; CNRS LTCI
Paris, France
first.last@telecom-paristech.fr

ABSTRACT

Nombreuses sont aujourd'hui les applications de veille sur *objets mobiles* : voitures, trains, avions, bateaux, personnes, ou, plus globalement, populations ou phénomènes tels que cyclones. De façon classique, cela exige la collecte de données à partir de réseaux de capteurs, d'analyse d'images ou de vidéos, ou l'utilisation de ressources spécifiques à une application cible. Nous montrons dans ce papier de démonstration comment le contenu Web peut être à la place exploité pour collecter des informations (trajectoires, métadonnées) concernant certains objets mobiles. Cependant, l'incertitude et les incohérences vont de pair avec les données Web. Nous développons ainsi une méthodologie pour l'estimation de l'incertitude et le filtrage des données extraites. En guise de démonstration, nous présentons sous forme d'une application Web un système construisant des trajectoires de bateaux à partir de données issues de réseaux sociaux, en présentant à l'utilisateur les trajectoires inférées, des méta-informations, ainsi que leurs niveaux d'incertitude.

1. INTRODUCTION

Moving objects and the Web. Consider the problem of tracking real-world *objects* such as cars, trains, aircrafts, ships, persons (e.g., celebrities), or, more broadly, populations or groups of humans, natural phenomena such as cyclones, epidemics. Such moving objects are characterized by timestamped location data and other meta-information such as name, size, maximum reachable speed, acceleration patterns, etc. The analysis and mining of spatio-temporal information about moving objects is common in a variety of applications, e.g., for pattern discovery [3, 6, 8, 13] or prediction of trajectories and locations [2, 7]. The overall goal may be to better understand certain natural phenomena, to improve city services, to regulate route traffic, etc. Currently used methods for tracking moving objects are often complex, mostly rely on application-specific resources and costly equipment (e.g., satellite or radar tracking of ships and aircrafts), and may require using individuals' private information, raising privacy concerns [10].

The Word Wide Web, on the other hand, is a huge source of public information about various real-world moving objects. Timestamped geographical data about the position of moving objects are disseminated on the Web, notably on location-based social networks or media sharing platforms. Social networking sites like Twitter and Facebook have the ability of recording the real-time location of the user posting a message, thanks to data from the GPS system, or mobile or wireless networks. In addition, these messages may also contain location information as free text. Thus, it is theoretically possible to obtain information about the user herself, or any moving object that the user is referring to in her message. Media on sharing platforms like Flickr and Instagram may be annotated with automatically acquired spatio-temporal information, such as the timestamp and location of a picture as added by modern digital cameras.

In addition, the Web also provides in a variety of online databases more general information about moving objects. For instance, data such as the usual residence of a given individual can often be found online, e.g., in Wikipedia or Yellow Pages services. Characteristics of particular flights and ships are available on specialized Web platforms. In this demonstration paper, we illustrate how to extrapolate on the information extracted from multiple Web sources in order to infer the locations of certain moving objects at given times, and to obtain general information about these. We then visualize these locations, together with hypothetical trajectories, on a map-based representation. This information is uncertain, however, and exhibits many inconsistencies. One of the challenges to overcome is to estimate the inherent reliability of that information.

We claim Web information can be used in a variety of settings where one would like to track moving objects. We illustrate with the applications of celebrity spotting and ship monitoring ; the sources, data, and scenario of the demonstration will focus on the latter.

Celebrity spotting. Journalists, paparazzi, fans, detectives, intelligence services, are routinely interested in gathering data and following the travels of given individuals, usually celebrities. These individuals may be active in social networks such as Twitter and Instagram, where they share geolocated data (or data tagged with location information) ; they may also be spotted in real life by users of media sharing platforms, who will upload geolocated pictures. Various Web sites, news articles, etc., provide additional meta-information. Exploiting this mass of information would provide a cost-effective and legal manner to reconstruct a meaningful trajectory.

Ship monitoring. Researchers in maritime traffic investigate the routes followed by different kinds of ships to propose traffic optimization methods, to predict pollution levels, or to prevent pirating actions. Though ships do broadcast information about their position using the AIS (Automatic Identification System) [11], this information is not made publicly available, and no historical log is kept. Information about the timestamped location of cruise ships, military vessels, tankers, etc., is common on Web sharing platforms such as Flickr and on other specialized Web sources (see Section 4). Integrating ship data from multiple Web sources also helps obtaining more complete and certain information about their characteristics.

Related work and contribution. Discovering user routines based on geographical data from social networks has been studied in [8], focusing on a specific type of moving object, and does not consider visualization aspects ; in addition, the authors do not deal with uncertainty in used Web information. Web information is uncertain because of imprecise and incomplete data. Moreover, according to [12] location data are inherently uncertain since one is never sure whether those locations are approximate or really precise. In this paper, we extract data about moving objects through keyword search over a set of Web sources. We estimate the amount of uncertainty in each location for all kinds of moving object based on two main criteria : *outliers* and *far-fetched trajectories*. We introduce another criterion, namely *on-land locations*, pertaining to the specific maritime traffic demonstration application. For each non-geographical piece of information, we consider and integrate multiple possible values from different Web sources. A computation of the most probable value can be done using truth finding algorithms [1, 5].

Outline. We present in Section 2 our Web extraction approach for gathering locations and general information about moving objects. Section 3 describes a method for evaluating the precision of obtained locations and for integrating uncertain attribute values. Section 4 introduces the maritime traffic application and our implementation. Finally, Section 5 details the demonstration scenario. A video accompanying this demonstration paper is available at `http://dbweb.enst.fr/ships.mpg`.

2. DATA EXTRACTION

We distinguish between two types of Web information about a moving object : *location data* and *general information*. A location refers to a particular object's position, whereas general information describes a specific characteristic of this object. We extract object information from Web sources through keyword search. That is, we consider a key phrase corresponding to the name of the moving object and crawl data we obtain from a set of Web sources (see Section 4 for the specific sources used for ship monitoring). For object locations, we focus on location-based platforms and social networks which provide geolocated data items.

Gathering general information. We collect general information about moving objects based on a *supervised* extraction over a fixed set of Web sources. The main intuition is that for many moving objects, e.g., ships, general information provided by a number of Web sources is structured into Web templates. This is particularly true for domain-specific resources. Inside this template, each particular characteristic has a meaningful label, with a value associated to it. We implement, based on such observation, an extraction process over these Web sources by using source-specific functionality for keyword search, and then crawling and parsing obtained HTML pages. Through hand-written schema mapping rules, we return data items in a global schema as a collection of attributes and corresponding values. Since we consider different sources, we can obtain multiple distinct values for the same attribute.

Location extraction. We extract locations of moving objects by searching Web data items such as pictures, posts, and tweets that have geographical information attached to them (either directly as semantic geolocation information, or as can be extracted by a gazetteer on tags and free text). This type of Web data of interest can be found on the majority of popular location-based networks and social Web platforms like Flickr, Instagram, Facebook, etc. A geolocated Web data item comes with geographical data (latitude and longitude), a date and additional meta information such as a title, a description, a set of tags, a user name, etc. As an example, picture geolocation is sometimes available as *Exif* automatically recorded by a digital camera at the time the picture was captured. Technically, we proceed as follows for the extraction. Given a moving object name, from a set of social Web sources we first look for geolocated data items which are relevant with respect to the given keyword. Then, for each data item we extract geographical data, dates and meta information.

3. UNCERTAINTY ESTIMATION

We evaluate in this section the amount of uncertainty in moving object data extracted from the Web. We first estimate the precision of geographical data according to three criteria. Then, we present the integration of general information from different sources.

3.1 Estimating Precision of Geographical Data

As already mentioned, we harness geolocated Web data items for computing the different locations, thereby hypothetical trajectories, of moving objects. Geographical information associated to their geolocated data items come with imprecisions, however. First, for various reasons a keyword extraction approach is imprecise. As a result, the search may return wrong or irrelevant results regarding the moving object of interest. For instance, when searching geolocated Web data items about a moving object O, one can get from a given Web source results related to another type of real-life things, e.g., a street with a similar name. Second, even if the results obtained really describe the object O, either the timestamp or the location information may be wrong (because of poorly configured software, purposedly introduced errors, ambiguous location names for gazetteers, etc.). We need an automated manner to detect these potential errors, and the resulting uncertainty on the data.

As a general framework, we estimate the precision of locations related to any O against two criteria. First, we detect outliers, that is, isolated locations, which represent locations with high probabilities to be impossible compared to other ones, that form a more consistent set of locations. Second, we evaluate the amount of imprecision in geographical data by analyzing whether two successive locations in a chronological sense form a realizable trajectory of O with respect to its maximum speed. For the purpose of our demonstration application, dealing with ships, we will also consider a third criterion which determines whether a location is in a water area or is, at least, near such an area. We next explain how we measure precision in each case.

Let $I_1, \ldots, I_j$ be a chronological sequence of distinct geolocated Web data items about the specific moving object O. A geolocated data item I_j is associated with a date $\mathrm{dat}(I_j)$, and location information $\mathrm{gd}(I_j)$. The latter is a pair $\mathrm{gd}(I_j) = (\varphi_j, \lambda_j)$ where φ_j and λ_j respectively represents the latitude and the longitude of a specific point on Earth. We use the simple point-location model of [12] and represent more formally a specific location of the moving object O as a couple $(\mathrm{gd}(I_j), \mathrm{dat}(I_j))$ where $\mathrm{gd}(I_j)$ is geographical coordinates and $\mathrm{dat}(I_j)$ is a date for all geolocated data item I_j. The set of different locations of O is thus $\langle(\mathrm{gd}(I_1), \mathrm{dat}(I_1)), \ldots, (\mathrm{gd}(I_j), \mathrm{dat}(I_j))\rangle$ with respect to data items $I_1, \ldots, I_j$.

Fix two locations $(\mathrm{gd}(I_i), \mathrm{dat}(I_i))$ and $(\mathrm{gd}(I_j), \mathrm{dat}(I_j))$. Necessarily, $i \leq j$ if and only if $\mathrm{dat}(I_i) \leq \mathrm{dat}(I_j)$. Given the non-planar shape of the Earth, the distance d_{ij} between these two locations of O is computed via the Haversine formula [9].

Detecting possible outliers. An outlier denotes a location far away from a set of locations, consistent with respect to a given time interval and maximum distance. The outlier, together with the set of consistent locations, describe the identical moving object O. We proceed as follows to detect outliers. We fix ε to be a maximum distance within a cluster. We say that a point is an outlier if its date falls into an interval where all other points are at distance ε of at least another point of the interval, while the outlier does not. Formally, for arbitrary $i < j$, we say that $(\mathrm{gd}(I_q), \mathrm{dat}(I_q))$ with $i < q < j$ is an outlier within $\Sigma = \{\, (\mathrm{gd}(I_k), \mathrm{dat}(I_k)) \mid i \leq k \leq j, k \neq q \,\}$ if :

– for all $(\mathrm{gd}(I_k), \mathrm{dat}(I_k)) \in \Sigma$, there exists $(\mathrm{gd}(I_{k'}), \mathrm{dat}(I_{k'})) \in \Sigma$ with $k \neq k'$ and $d_{kk'} \leq \varepsilon$;

– for all $(\mathrm{gd}(I_k), \mathrm{dat}(I_k)) \in \Sigma$, $d_{kq} > \varepsilon$.

To define ε and the length of the time interval, we suppose given a number m of expected clusters such that the number of locations is equally distributed through the clusters. That is, we have $\frac{n}{m}$ points within each cluster Σ for n input locations (a slight overflow in one cluster is tolerated if a uniform distribution can not be respected). Then, we divide the overall interval $[\![1, n]\!]$ into subintervals of length $\frac{n}{m}$. For any such subinterval $[\![i, j]\!]$ of length at least $\frac{n}{m}$, we set ε as being the average of the distances $d_{kk'}$ for all $i \leq k < j$ and $k' = k+1$.

Far-fetched trajectories regarding reference speed. A possible itinerary of the moving object O maps to a connected set of chronologically ordered locations. A trajectory may be far-fetched, i.e., unreasonable, if reaching one location from a previous one is impossible when we consider the reference speed of O. We are not interesting here by the outlier locations of O. Let V be the reference speed value of O induced from gathered general information. Given two consecutive locations $(\mathrm{gd}(I_i), \mathrm{dat}(I_i))$ and $(\mathrm{gd}(I_j), \mathrm{dat}(I_j))$ with $j = i+1$, we verify whether the following inequality holds : $d_{ij} \leq V \times (\mathrm{dat}(I_j) - \mathrm{dat}(I_i))$.

If not, we are in the presence of an impossible trajectory. At least one of these two locations is wrong. We do not know in advance which one, and it just participates in a confidence level computation.

Detecting on-land locations. Let us now introduce the detection of on-land locations of a moving object. This measure pertains to our maritime traffic application in which we are mostly interested in locations falling in water areas. Other applications will have similar application-specific ways of detecting impossible points.

We define a location on land as a point on Earth which is out of water areas like seas, oceans, rivers, lakes, etc. Data about all water regions on Earth can be found on the Web platform Natural Earth[1] in the form of *multi-polygons* for lakes, seas and oceans, and *polylines* for rivers. Based on these shapes and the *ray-casting algorithm*, we check whether a given location $(\mathrm{gd}(I_j),\ \mathrm{dat}(I_j))$ of the moving object O (here typically a ship) falls within one of the considered polygons or polylines. We estimate all on-land locations for O in this manner. Observe that some of these locations on land could be relevant for our application. In particular, locations on land, e.g., ports, that are close to water areas. To account for those kinds of interesting locations on land, we introduce a *tolerance factor* by considering the disc with radius of x and centered on a location. In the demonstration application, we set x to 0.1 degree of latitude/longitude. Given that, we find on-land locations whose disc, w.r.t. the tolerance factor, intersects one of the polygons or polylines in some points in a water area. We consider finally the overall set of the on-land locations as being those that do not satisfy this condition.

3.2 Integrating Uncertain Attribute Values

We do not only extract from the Web geographical information. We also collect general information about the moving object O in the form of attributes and corresponding values. Attributes are distinguished by meaningful labels specific to the individual sources. In general, Web sources have different level of completeness in terms of the data they provide. In addition, some of them can provide conflicting information, i.e., there can be multiple possible attribute values for a given attribute, coming from different Web sources.

As general information comes from multiple sources, we need to integrate them in order to provide to the user a unique global view. In this integration process, we have to deal with the uncertainty that is inherent to the Web, but also that results from contradictions. We integrate general information about O from multiple Web sources by first matching values of the same attribute provided by distinct sources, using a manually constructed schema mapping across sources, and then by merging identical values. When a conflict occurs, we consider the value provided by the majority of sources as the most reliable one, but we keep all different values, as will be clear in the demonstration. This process for choosing the most probable values among conflicting ones corresponds to a voting approach. More elaborate voting strategies can be used, such as those given in [5].

4. MARITIME TRAFFIC APPLICATION

Use case. The use case of our demonstration is the monitoring of ships. We rely on Flickr for collecting a large amount of geographical information about the different locations of a given ship. Flickr provides an easy-to-use API[2], with a set of predefined functions, e.g., `flickr.photos.search`, for extracting all pictures (each with a unique identifier), together with necessary meta-data including geographical coordinates and dates, whose title, description, or tags contains a certain keyword given in input. The extraction process on top of the Flickr platform is automated using API Blender [4], an open-source library facilitating interactions with the Flickr API. As for the general information on ships, in particular details about their specifications, we integrated information from GrossTonnage[3], Marinetraffic[4], ShippingExplorer[5], ShipSpotting[6], and Wikipedia[7]. These sources contain general information about various types of ships. The purpose of the first three is to gather data about objects in the maritime domain, especially vessels. However, excepting Marinetraffic that provides partial information under an API, these Web sources do not provide a way to extract specific information from their platforms, and need to be crawled.

We encode extracted Web data using JSON. We show in Figure 1 a JSON code excerpting a geolocated picture about the ship "Liberty of the Seas". This picture, identified by "8442802776", has a date and geographical data mapping to "2013-01-28 19 :53 :50" and "[25.865209, -80.031677]", respectively. It is provided by the user "Michael Bentley". We compute a trust score for this user as the percentage of *good* locations, w.r.t. the three criteria in Section 3, given its entire uploaded pictures for this ship.

Implementation. Our demonstration system has the form of a Web application with a map displaying ship locations. We implemented the full system using HTML, CSS, and JavaScript on the client-side, and Python on the server-side. The Projection of raw geographical data onto a map uses the popular Google Maps JavaScript API. Finally, features such as filtering options are performed using the jQuery JavaScript library.

5. DEMONSTRATION SCENARIO

[1] `http://www.naturalearthdata.com/`

[2] `https://www.flickr.com/services/api/`

[3] `http://grosstonnage.com/`

[4] `http://www.marinetraffic.com/fr/ais/home/`

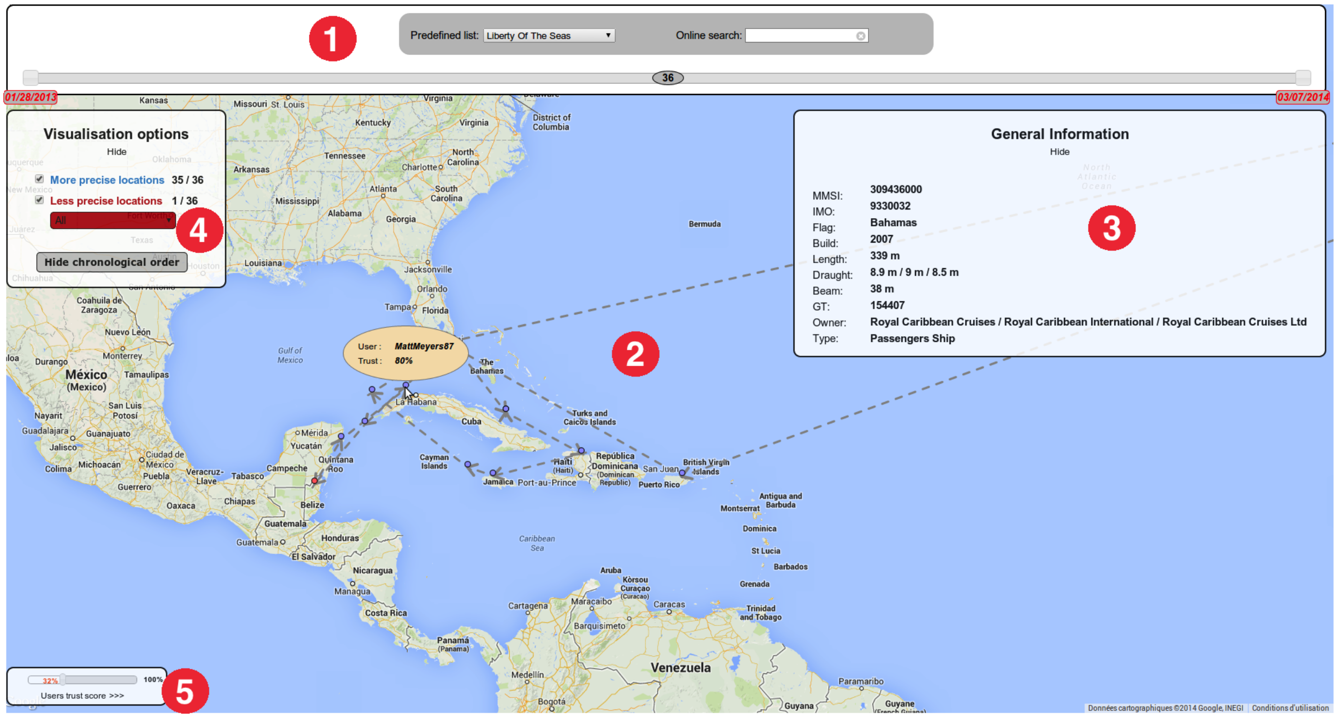
[5] `http://www.shippingexplorer.net/en`

[6] `http://shipspotting.com/`

[7] `http://en.wikipedia.org/wiki/Main_Page`

```
[["8442802776", {"username": "Michael Bentley", "userID": "35456872@N00"}, "2013-01-28 19:53:50", [18, {"source"
    : "http://farm9.staticflickr.com/8231/8442802776_6bebdf9ff1.jpg"}], "tagged", [25.865209, -80.031677] ]...]
```

Figure 1: A geolocated picture from Flickr



Predefined list: Liberty Of The Seas

Online search:

Visualisation options

Hide

☑ More precise locations 35 / 36

☑ Less precise locations 1 / 36

All

Hide chronological order

General Information

Hide

MMSI: 309436000

IMO: 9330032

Flag: Bahamas

Build: 2007

Length: 339 m

Draught: 8.9 m / 9 m / 8.5 m

Beam: 38 m

GT: 154407

Owner: Royal Caribbean Cruises / Royal Caribbean International / Royal Caribbean Cruises Ltd

Type: Passengers Ship

User : MattMeyers87

Trust : 80%

32% 100%

Users trust score >>>

Figure 2: Main interface of our system

Interface. To interact with the system, a given user can either choose a ship name in a predefined list with locally saved data, or trigger an on-line search over the considered Web sources by providing a keyword (see Region 1 in Figure 2). For the live Web search, the user can restrict the proposed set of sources for the extraction of the general information about the requested ship. The system will integrate obtained information when multiple sources are involved. Once information is obtained in local or from the Web, the different locations are displayed on the map and the general information is shown (Regions 2 and 3 in Figure 2).

Ship positions are divided by default into two categories with different colors. Blue points on the map correspond to locations with high precision, whereas red points come with less precision, i.e., these are less reliable. As for the general information, we only show the most probable value for each attribute. The user has, however, the possibility to see details about possible other values by hovering the mouse over each attribute label. The user can restrict the visualization to ship positions in a given time interval with the slider at the top of the interface. Over this slider, we have the total number of mapping locations. More advanced visualization options are available, as shown in Regions 4 and 5 in Figure 2 : The user can filter locations with high or low precision. For low precision locations, she can focus either on those on the land, outliers, or locations leading to impossible trajectories. Finally, the user can visualize hypothetical itineraries, filter users according to given trust scores, or restrict to specific users.

Example interaction. An expert in the maritime domain would like to acquire new ships with specific characteristics and history for business purposes. A company sells vessels which may correspond to her needs. Two particular passenger ships "Liberty of the seas" and "Costa Serena" are of interest. Thus, she decides to verify from various Web sources whether the details given by the seller are correct before making a definitive choice. To do so, she uses our maritime traffic application which already holds information about "Liberty of the seas" and "Costa Serena" in local. The user selects the first one and obtains the map view of locations. She primarily overviews the general information about the ship, and observes conflicting values for its draught and its owner. The user thus checks the values of these two attributes, as given by her most trusted Web sources. These values seem to be consistent with the seller's data after verification. The user remembers that she is very interested in positions of "Liberty of the seas" at some periods of the year (January to March and August to November). She filters positions corresponding to these date intervals with the slider. Surprisingly, the user remarks that the ship was near the Caribbean Sea and the Mediterranean Sea in these times. These information contradict the seller who had stated that the ship has never left Europe. To obtain more insight about the journeys, the user triggers the view of hypothetical trajectories. She examines the choice list of less precise locations to understand why some points are incorrect. She concludes that all among them are on-land and indeed invalid. Finally, she removes these kinds of locations from the map, which confirms that ship routes mostly cover two main regions.

The user pursues explorations by considering "Costa Serena" now. She only focuses on its positions and past destinations. She notes that less precise locations make the visualization cumbersome with no clear overview on routes. Therefore, the user filters one by one each type of less precise locations for explanations. For instance, she picks on-land locations and notices that all of them are located on *Corsica*, an island which contains a region named "Costa Serena" – she learns this information by clicking on an on-land location and reading the corresponding Flickr page. Observing that providers of less precise locations have trust scores below 100%, the user sets the minimum trust to 80%. Finally, she refines remaining locations w.r.t. given intervals of dates, comparing with data from the selling company, and can therefore make an informed purchase decision.

A video presenting our demonstration scenario is available at `http://dbweb.enst.fr/ships.mpg`.

6. REFERENCES

[1] M. L. Ba, S. Montenez, R. Tang, and T. Abdessalem. Integration of web sources under uncertainty and dependencies using probabilistic XML. In *UnCrowd*, 2014.

[2] G. K. D. De Vries and M. Van Someren. Machine Learning for Vessel Trajectories Using Compression, Alignments and Domain Knowledge. *Expert Syst. Appl.*, 39(18), 2012.

[3] B. Furletti, P. Cintia, C. Renso, and L. Spinsanti. Inferring human activities from GPS tracks. In *UrbComp*, 2013.

[4] G. Gouriten and P. Senellart. API Blender: A uniform interface to social platform APIs. In *WWW*, 2012. Dev. track.

[5] X. Li, X. L. Dong, K. Lyons, W. Meng, and D. Srivastava. Truth finding on the deep Web : is the problem solved ? *PVLDB*, 6(2), 2012.

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

[7] M. Morzy. Mining Frequent Trajectories of Moving Objects for Location Prediction. In *MLDM*, 2007.

[8] F. Pianese, X. An, F. Kawsar, and H. Ishizuka. Discovering and predicting user routines by differential analysis of social network traces. In *WoWMoM*, 2013.

[9] R. W. Sinnott. Virtues of the Haversine. *Sky and Telescope*, 68(2), 1984.

[10] D. Soper. Is Human Mobility Tracking a Good Idea ? *Commun. ACM*, 55(4), 2012.

[11] H. Taka, D. Shibata, M. Wada, H. Matsumoto, and K. Hatanaka. Construction of a marine traffic monitoring system around the world. In *OCEANS*, 2013.

[12] O. Wolfson. Moving Objects Information Management : The Database Challenge. In *NGITS*, 2002.

[13] H. Yuan, Y. Qian, R. Yang, and M. Ren. Human mobility discovering and movement intention detection with GPS trajectories. *Decision Support Systems*, 2013.