2

Trajectory Collection and Reconstruction

Gerasimos Marketos, Maria Luisa Damiani, Nikos Pelekis,
Yannis Theodoridis, Zhixian Yan

2.1 Introduction

The research area of trajectory databases has addressed the need for representing movements of objects (i.e., trajectories) in databases in order to perform ad-hoc querying and analysis on them. During the last decade, there has been a lot of research ranging from data models and query languages to implementation aspects, such as efficient indexing, query processing and optimization techniques.

This chapter covers aspects related to data collection and handling so as to feed trajectory databases with appropriate data. We will also focus on the step *trajectory reconstruction* of the *Geographic Privacy-aware KDD process* (illustrated in Figure 2.1) emerged from the GeoPKDD project which proposed some solid theoretical foundations at an appropriate level of abstraction to deal with traces and trajectories of moving objects aiming at serving real world applications. This process consists of a set of techniques and methodologies that are applicable on mobility data and are organized in some well-defined and individual steps that have a clear target: to extract user-consumable forms of knowledge from large amounts of raw geographic data referenced in space and in time. However, when mobility data is about individuals, data collection is subject to privacy regulations and restrictions. To enable privacy-aware collection of position data, a complementary class of techniques are used, known as *location PETs* (privacy-enhancing technologies).

This KDD process can be applied to heterogeneous sources of mobility data. The cellphone icon that is illustrated in Figure 2.1 could represent various data sets coming from various devices. In Section 2.2, we present such sources.

Before applying trajectory reconstruction techniques we may need to perform some basic trajectory preprocessing. This may include parameterized trajectory compression (so as to discard unnecessary details and concurrently

keep informative abstractions of the portions of the trajectories transmitted so far), as well as techniques to handle missing/erroneous values. Moreover, to deal with moving object applications that are restricted to some network, map-matched trajectories may be needed. In other words, we may need the specific trajectory points and portions to correspond to valid network paths. This may include for example, performing pre-processing or post-processing tasks that do not violate the validity of trajectories in terms of the real underlying network. We describe this kind of tasks as trajectory data handling and we present them in Section 2.3.

In Section 2.4, we present trajectory reconstruction techniques for transforming sequences of raw sample points into meaningful trajectories and store them into trajectory databases. The reconstructed trajectories can be either semantic-free (raw trajectories) that just represent the movement of an object or semantically enriched, containing information about the nature of the movement.

Section 2.5 presents techniques for the privacy-preserving collection of trajectory data.

2.2 Tracking Trajectory Data

In this section, we present some technologies that can be used for tracking trajectories of moving objects. More specifically, these technologies provide us access to position data that may represent an incomplete, partial or vague representation of the real movement of moving objects but with the appropriate handling techniques (Section 2.3) can lead to the reconstruction of trajectories (Section 2.4).

GPS Data GPS is the fully-functional satellite navigation system that utilizes more than two dozen satellites. It broadcasts precise timing signals by radio to GPS receivers, allowing them to accurately determine their location (longitude, latitude, and altitude) in any weather, day or night, anywhere on Earth. A GPS receiver calculates its position by precisely timing the signals sent by GPS satellites high above the Earth. Each satellite continually transmits messages that include:

- the time the message was transmitted
- precise positioning information
- the general system health and rough orbits of all GPS satellites

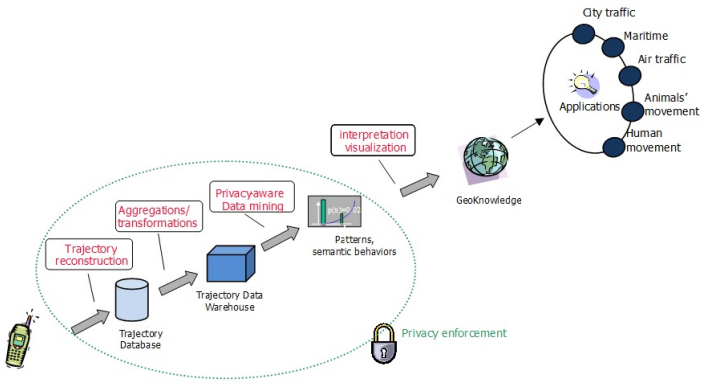


Figure 2.1 The big picture of moving object data management, warehousing and mining concepts.

The receiver computes the distance to each satellite by using the messages it receives to determine the transit time of each message. These distances along with the satellites' locations are used to compute the position of the receiver. This position is then displayed, perhaps with a moving map display or latitude and longitude; elevation information may be included. Many GPS enabled devices show derived information such as direction and speed, calculated from position changes. GPS enabled devices provide us with all the required information for trajectory tracking. They give us access in accurate time stamped locations for each tracked moving point.

GSM Data GSM is the most popular standard for mobile phones in the world, nowadays used by over 1.5 billion people across more than 210 countries and territories. The ubiquity of the GSM standard makes international roaming very common between mobile phone operators, enabling subscribers to use their phones in many parts of the world. GSM networks consist of a numbers of base stations each responsible for a particular spatial area (known as cell'). Hence, for each GSM-enabled device we can collect information about the base stations it was served at different timestamps, and as such, assume its movement.

A GSM-enabled device can be tracked by collecting all the communication signals transmitted (cell, signal strength) between this device and the network infrastructure or by studying the log of the out-coming calls (UserID, data and time of the call, duration of the call, the cell where the call began, the cell where the call finished). However, in both levels the accuracy of trajectories that can

be collected is very low since the most detailed level of available information is the network cell and not a spatial point.

Bluetooth Data The movement of a Bluetooth device within an area can be tracked by considering the distances of the device from Bluetooth receivers and using trilateration approaches. The distance of a Bluetooth device from a specific receiver can be calculated using techniques that consider signal levels.

The disadvantage of this technique is that it can be mainly used for in-door tracking of objects as Bluetooth receivers cover a limited area and they cannot really be used for outdoor object tracking.

RFID Data The purpose of an RFID system is to enable data to be transmitted by a portable device, called a tag, which is read by an RFID reader and processed according to the needs of a particular application. A typical RFID tag consists of a microchip attached to a radio antenna mounted on a substrate. A typical chip can store as much as 2 kilobytes of data. A reader is needed to retrieve the data stored on an RFID tag. A typical reader is a device that has one or more antennas that emit radio waves and receive signals back from the tag. The data transmitted by the tag may provide identification or location information, or specifics about the product tagged, such as price, color, date of purchase, etc. Same as in Bluetooth technology, RFID readers can locate tags within a limited area so it is hard to apply this technology for outdoor tracking of moving objects.

2.3 Handling Trajectory Data

Real-life trajectory data, collected using the technologies previously presented, are not really readily used for analysis purposes. In this section, we elaborate on various approaches for handling trajectory as a necessary step for identifying *clean* (i.e. without noise), *accurate* (i.e. map-matched), and *compressed* (i.e. compact) trajectories, from the original sequence of spatio-temporal positions (e.g., GPS records) of the moving objects.

2.3.1 Data Cleaning

Data sets collected by mobile sensors are often imprecise either unintentionally, due to limitations of positioning systems (e.g., inaccurate GPS measurement and sampling errors, signal loss, battery running out) or intentionally so

as to protect individuals' privacy (e.g., people may expose an approximation of their positions).

In case of unintentional (GPS) errors, trajectory cleaning (i.e. removing errors) is an important step in the procedure of constructing meaningful raw trajectories from the GPS feeds. Generally speaking, two types of GPS errors can be identified: *systematic* errors, due to system's limitations, and *random* errors, due to external reasons. Systematic errors can be caused by horizontal dilution of position (HDOP) due to the low number of available satellites, while random errors are small errors up to ± 15 meters caused by the satellite orbit, atmospheric and ionospheric effects, and receiver issues. We should notice here that errors are related to the spatial positions and not to the temporal aspect of mobility as it is considered highly precise.

In order to identify systematic errors, researchers may resort to visual inspection in case of small data sets. For that reason, we could use a filtering method that filters noisy positions by taking advantage of the maximum allowed speed of a moving object. This threshold/parameter is used in order to determine whether a reported position from the GPS stream must be considered as noise and consequently discarded, or kept as a normal record.

On the other hand, random errors are small distortions from the true values. Their influence is reduced by smoothing methods. In the literature, different approaches can be found based on Gaussian kernels, where a smoothed spatial position is the weighted local regression based on past and future positions within a sliding time window considering the weight as a Gaussian kernel function, and Kalman filter, which uses measurements observed over time (the positions coming in the GPS receiver), and predicts positions that tend to be closer to the true values of the measurements.

2.3.2 Map Matching

The previous trajectory cleaning methods are designed for objects moving without any constraint in their movement. However, real-world applications usually consider objects that are restricted to move within a given spatial network that is represented as a graph (e.g., road/railway network) (you can find more information about this topic on Chapter 3). Other applications may consider spatio-temporal constraints (e.g., a pedestrian cannot walk at a speed above a certain limit, usually bats don't fly during daytime).

For network-constrained trajectories, the map-matching approach refers to the mapping of a trajectory to the edges and nodes of the network. More precisely, the general idea is the replacement of each position of the original trajectory by the point on the network that is the most likely position of the mov-

ing object. From a computational point of view, map-matching methods can be categorized to online (processing streams of new positions in real time) or offline (when all positions are available), while both groups can be further classified as *geometric*, *topological*, or *hybrid* methods.

Geometric methods take into consideration the underlying road network and various distance measures to determine the actual traveled roads. These distance measurements can be point-to-point (e.g., Euclidian distance), point-to-curve (e.g., perpendicular distance), or curve-to-curve (e.g., Fréchet distance). For instance, Dijkstra's shortest path algorithm can be used to determine the distance between a trajectory and a sequence of arcs on a map. The route with the smallest distance from the initial trajectory is taken as the map-matched trajectory. For instance, Figure 2.2 illustrates such a methodology: for every point P_i, given that point P_{i-1} has already been matched to an edge, the adjacent edges to this edge are the candidate edges to be matched to P_i and they are evaluated, as illustrated in Figure 2.2. In this example, P_{i-1} is matched to edge c_3, hence c_1, c_2 and c_3, are the candidate edges for point P_i. Two measures are used for choosing among the candidate edges that are based on similarity and orientation criteria. The higher the sum s of these measures is, the better the match to this edge is. If the projection of the current point on the candidate edges does not lie in-between the end points of any of these edges, the algorithm does not proceed to the next point. Instead, the nearest edge of the candidates is set as part of the trajectory and then the next set of candidate edges is evaluated. On the contrary to geometric approaches, the topological

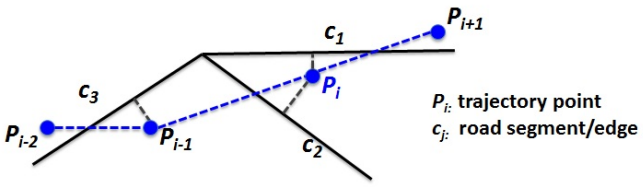


Figure 2.2 Applying map matching.

approaches account for the connectivity and contiguity of the road network without assuming any knowledge of the expected traveling route and the speed or heading information supplied by the GPS.

More recent map-matching methods deal with the problematic case where GPS data are arriving with low sampling rate (e.g., one point every two minutes) and high noise. These new methods employ both distance and topology and aim to align an entire trajectory with the road network. In some cases,

not only distance and topology are used but also Hidden Markov Model approaches to find the most likely road route corresponding to a sequence of positions.

The various proposals usually include several post-processing techniques to calibrate and correct the initial matching results. Obviously this worsens the cost/efficiency of the algorithm. This is an important issue that should be addressed by future research.

2.3.3 Data Compression

Trajectory data in applications grow progressively and intensively as the tracking time goes by. Such huge amounts of data raise storage, transmission, computation, and display challenges. Therefore, trajectory data compression is an essential task of trajectory reconstruction. The bibliography in this area usually assumes that the objectives of trajectory compression are: (1) to reduce the size of the data set, (2) the reduced data set should allow computations of acceptable/low complexity, and (3) a trajectory from the reduced data set should not deviate from the original one by more than a given threshold.

From a geometric perspective, compression techniques exploit on line simplification algorithms that remove positions from a trajectory without warping the trend of the trajectory or distorting the database. In general, trajectory compression algorithms can be classified into four categories: *top-down*, *bottom-up*, *sliding window*, and *opening window*. The top-down algorithm recursively splits the sequence of positions and only keeps the key (*representative*) positions in each sub-sequence, i.e. the ones that lie far from the line that would result if these points were removed. A classical top-down method is the Douglas-Peucker (DP) algorithm, with many subsequent extensions. The bottom-up algorithm starts from the finest possible representation, and merges the successive points until some halting conditions are met. Sliding window methods compress data in a fixed window size; whilst open window methods use a dynamic and flexible data segment size.

For instance, the *Top-Down Time Ratio* (TD-TR) and *Open Window Time Ratio* (OPW-TR) algorithms have been proposed for the compression of spatio-temporal data. The TD-TR approach uses the DP algorithm and, moreover, takes the time into account. In particular, it replaces the Euclidean distance used in DP by a time-aware one, called Synchronous Euclidean Distance (SED) as illustrated in Figure 2.3. In this example, let P_b be the currently examined point against line P_1P_n. The DP approach uses the perpendicular distance of P_b to P_1P_n, while the TD-TR uses the distance of P_b to $(P')_b$ (i.e. the SED). The coordinates of point P'_b are calculated using linear interpolation. The OPW-TR

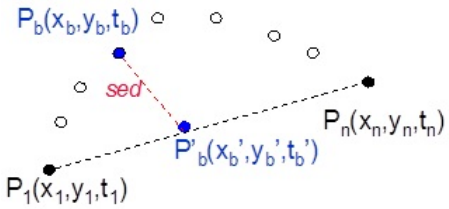


Figure 2.3 Using SED.

algorithm works as follows. Initially, it defines a line segment between the first and the third data point. If the SED from each internal point to the segment is not greater than a given threshold, the algorithm moves the end point of the segment one position up in the sequence. When the threshold is exceeded, the data point that causes the threshold excess or its precedent is defined as the end position of the current segment and the start position of a new one. As long as new positions arrive, the method continues as described.

Two other interesting algorithms in the bibliography are the Thresholds and STTrace, appropriate for online trajectory data compression. The algorithms use the coordinates, speed, and orientation of the current position in order to calculate a safe area where the next position might be located. If the next incoming position lies in the calculated safe area, it can be ignored. There are two options for the definition of the safe area. It is either calculated by using the last position, whether it has been previously ignored or not, or by using the last chosen position. In order to achieve better results, a combination of the two algorithms is also proposed. Both areas are calculated, but only their intersection is defined as the safe area.

These trajectory compression approaches are primarily based on the extension of geometric methods like the DP algorithm. However, they are not suitable for network constrained trajectories. Therefore, recent works proposed another kind of trajectory compression models that make use of the underlying road network. By map-matching, trajectories can be reconstructed (or represented) by only the matched road segments, without the need for keeping the original movement points.

2.4 Reconstructing Trajectories

Chapter 1 introduced the differentiation between raw and semantically enriched trajectories. Here we present reconstruction techniques for both types.

Trajectory reconstruction refers to the task of transforming raw spatio-temporal positions into meaningful trajectories. An interesting note here is that different applications may need different trajectories. For instance, there may be a considerable difference on the semantic definition of a trajectory given by a traffic analyst and, on the other hand, a logistics manager. Let us consider a fleet of trucks moving in a city and delivering goods in various locations. The logistic manager may consider, for each truck, a number of different trajectories (e.g., between the different delivery points) while the traffic analyst may consider a single trajectory for the whole day. Thus, in order to satisfy these two, quite different in semantics, requirements we would have to retrieve raw spatio-temporal position data from a common repository and, then, execute two different reconstruction tasks so as to produce trajectories that are semantically compliant to each domain. For instance, Figure 2.4a illustrates a raw data set of spatio-temporal positions. Different needs may result in different set of reconstructed trajectories Figure 2.4b-d, respectively). Recalling the previous example of the truck data set, let us consider Figure 2.4b and c that illustrate the reconstructed trajectories for the logistic manager and for the traffic manager respectively. Another example of trajectory reconstruction is presented in Figure 2.4d which considers a compressed trajectory of the movement. The exact number of reconstructed trajectories depends on the different semantic definitions that can be given to a trajectory. In this section, we present reconstruction techniques that can be used to produce either raw or semantically enriched trajectories.

Reconstructing Raw Trajectories Collected raw data represent spatio-temporal locations (Figure 2.5a). Apart from storing these raw data, we are also interested in reconstructing trajectories (Figure 2.5b). The so-called *trajectory reconstruction* task is not a straightforward procedure. Having in mind that raw points arrive in bulk sets, we need a filter that decides if the new series of data is to be *appended* to an existing trajectory or not.

The process of algorithm reconstruction needs a method for determining different trajectories, which should be applied on raw positions. Taking into consideration that the notion of trajectory cannot be the same in every application due to the fact that different requirements and semantics arise, some generic trajectory reconstruction parameters can be:

- *Temporal gap between trajectories*: the maximum allowed time interval between two consecutive spatio-temporal positions of the same trajectory for a single moving object (case *a* in Figure 2.5a).
- *Spatial gap between trajectories*: the maximum allowed distance in 2D plane

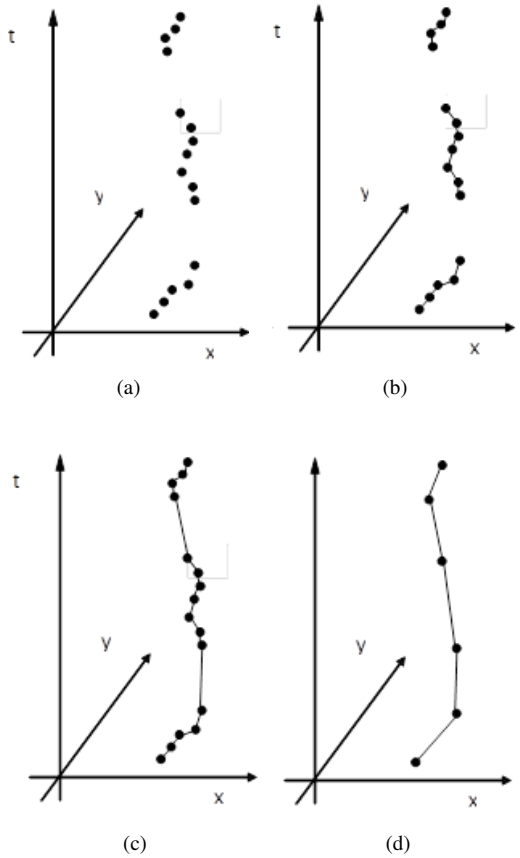
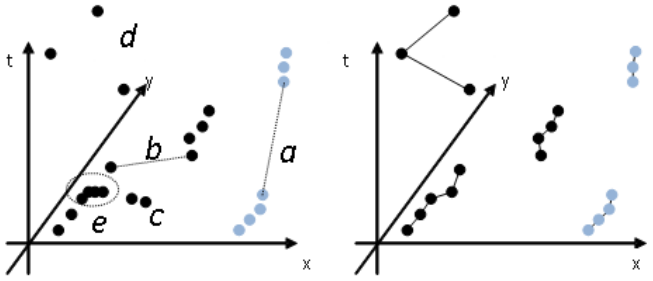


Figure 2.4 Three different trajectory reconstruction approaches (b, c, d) for a raw data set (a).

between two consecutive spatio-temporal positions of the same trajectory (case *b* in Figure 2.5a).

- *Maximum speed*: the maximum allowed speed of a moving object, used to determine noisy spatio-temporal positions (case *c* in Figure 2.5a).
- *Maximum noise duration*: the maximum duration of a noisy part of a trajectory so as to consider creating a new trajectory containing this part (case *d* in Figure 2.5a).
- *Tolerance distance*: the maximum distance between two consecutive spatio-temporal positions of the same object in order for the object to be considered as stationary (case *e* in Figure 2.5a).



(a) (b)

Figure 2.5 (a) raw locations, (b) reconstructed raw trajectories.

Reconstructing Semantic Trajectories Raw trajectories contain only spatio-temporal positions $\langle x, y, t \rangle$, which are insufficient for building meaningful trajectory applications. Therefore, researchers have proposed to reconstruct trajectories from the low-level collected data (e.g., GPS records, movement tracks) to high-level data abstractions, thus building semantic trajectories. The idea of semantic trajectories is to encode meaningful geo-locations/geo-objects (e.g., points of interest like a shopping mall, roads) into the raw spatio-temporal tracks; additional semantic annotations (e.g., trajectory behaviors like traveling in Paris, walking on Avenue des Champs-Elyses, taking Metro 3, shopping in a supermarket) are attached to the semantic trajectories.

Figure 2.6 briefly presents the main procedure of reconstructing such semantic trajectories from the raw GPS alike mobility records. From the initial GPS records, we can compute the trajectory episodes (e.g., stops, moves that are largely used in the literature to understand the structure of trajectories, presented in Chapter 1); afterward, a couple of dedicated annotation algorithms are provided for enriching trajectories using additional geo-objects and semantic tags. There are four main technical components for constructing such semantic trajectories, as follows:

- *Building trajectory episodes*: The aim is to build trajectory episodes to further understand the inner-structure of each individual raw trajectory. Trajectory episode is a sub-sequence of the raw trajectory. Trajectory data points inside one episode is more or less homogenous (e.g., staying in the same place, having the same travel speed), whilst data points in two neighboring episodes are unrelated. There are different kinds of episodes, such as Begin, End, Stop, and Move. In addition to these four types of episodes, additional

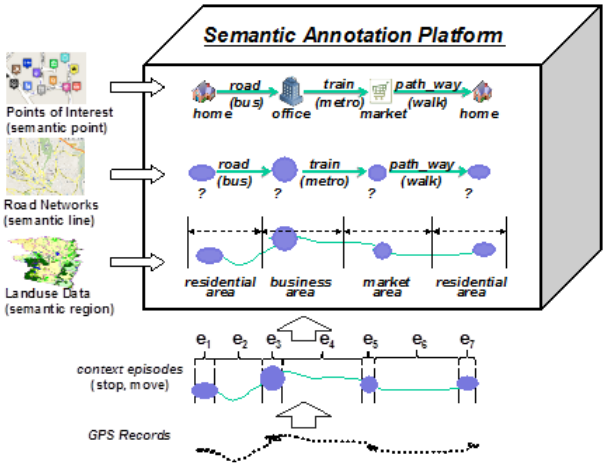


Figure 2.6 Annotation for semantic trajectories.

episode can be further designed according to the application scenarios, e.g., specific episode for representing congestions in traffic. The core issue here is to design efficient and robust trajectory segmentation algorithms to find these meaningful episodes. A couple of trajectory segmentation algorithms are proposed for building trajectory episodes, such as velocity, density, orientation, and even time-series based segmentation methods.

- *Annotating trajectory with regions*: This component enables annotation of trajectories with meaningful geographic or application domain sources of semantic regions. It does so by computing topological correlations between trajectories and third party data sources containing geo-objects of regions (called regions of interest or ROI). We need to design a spatial join algorithm, which can work for both regular regions (e.g., 100m $\times$ 100m grid-based land use data) and irregular regions (e.g., regions with free-style shapes like EPFL Rolex Learning Center).
- *Annotating trajectory with lines*: This component annotates trajectories with lines of interest (LOI) like road networks and considers variations present in heterogeneous trajectories (e.g., vehicles run on road networks, while human trajectories use a combination of transport networks and walk-ways etc.). Given data sources of different forms of road networks, the purpose is to identify *correct* road segments as well as infer the transportation modes such as "walking", "cycling", and "public transportation" like metro and bus. Thus, the algorithms in this component include two major parts: the first part is designing/reusing a global map matching algorithm to identify

the correct road segments for the move episodes of a trajectory, and the second one is inferring the transportation modes that the moving objects/people used during their moves.

- *Annotating trajectory with points*: This component annotates the Stop episodes in trajectory using information about suitable points of interest (POIs). Examples of POI are "restaurants", "bars", "shops", "movie theaters", etc. For scarcely populated landscapes, it is relatively trivial to identify the objective of a stop (e.g., petrol pump on a high-way, back home in a very sparse residential area). However, densely populated urban areas bring many different types of candidate POIs for a trajectory stop. The problem of inferring stop behaviors using POIs becomes challenging. Further, low GPS sampling rate due to battery outage and GPS signal losses makes the problem more intricate. Recently, a HMM (Hidden Markov Model) based inference algorithm has been designed to extract the underlying stop behaviors in the trajectory. In this algorithm, the location of individual trajectory stop is modeled as a model observation, whilst the POI category is considered as the hidden state that needs to be extracted.

2.5 Protecting the Privacy of Individuals' Positions

This section overviews techniques which aim at protecting users' privacy during the data collection process. The concern for privacy stems from the fact that whenever postion refers to individuals, position is qualified as personal data, while collecting personal data is restricted by privacy norms and law in several countries worldwide. In particular semantic trajectories magnify the risk for privacy because behavior information on individuals is explicitly extracted and represented in a machine-readable form, therefore can be used within information processing applications and easily unfolded to third parties. Though fundamental, privacy regulations are not capable of preventing malicious and curious parties from improperly accessing and use collected data. This instead is the goal of location PETs (Privacy-Enhancing Technologies). In general, location PETs can be applied at two different stages:

1 Before position data are collected. In this case the goal of location PETs is to prevent mobility data collectors from obtaining the exact location and trace of individuals, everytime and everywhere. Because these techniques are applied on the fly, we refer to this form of protection as *on-line location privacy*

2 After position data are collected and trajectories reconstructed. The goal of location PETs is to shape trajectory data in a way that the data set can be published or released to some other party without incurring privacy violations. We refer to this as *off-line location privacy*.

Off-line and on-line location privacy present different requirements which call for different solutions. In particular, the solutions for the on-line protection of location privacy have to deal with incomplete knowledge of the individuals' trajectories (usually only the current and past positions are known); moreover techniques must be efficient so as not to compromise the effectiveness of data collection. In what follows, we survey major paradigms supporting on-line location privacy while techniques for off-line location privacy will be presented later on in Chapter 9.

2.5.1 Online Location Privacy

Research on position privacy took off early last decade with the emergence of mobile applications enabling the tracking of moving objects, e.g., the vehicles monitored by a fleet management system and location-based services (LBS), e.g., search of points of interests nearby. These applications typically rely on a client-server architecture: the position is collected by mobile devices (the clients) and conveyed to a server handled by a service provider. In this scenario, service providers are in the position of collecting large amounts of position data, therefore if they are irrespective of users rights and requirements or, simply, the collected data are stolen, users' privacy is at stake. Commonly location PETs seek to limit the transmission of either accurate or explicit location information to service providers. These techniques can be further classified based on the information to be protected, i.e. the privacy goals. In particular, we distinguish three main goals: *identity privacy*, *location privacy* and *semantic location privacy*. In what follows we survey representative location PETs addressing these goals.

Identity Privacy Identity privacy techniques are conceived to forestall the re-identification of seemingly anonymous users, based on position information. For example, consider the case in which an LBS is offered to the members of a community potentially subject to discrimination, e.g., the gay community, and assume users to interact with the system through pseudo-identifiers. Unfortunately simply stripping off users' identifiers is not sufficient to ensure anonymity, because the service provider can draw identities from trajectory information, e.g., if a user requests the service from a certain place early in the

morning, it is likely that such a place is his or her home and thus the user can be easily re-identified through a white pages service. While we refer the reader to the literature for a survey of identity privacy techniques we limit ourselves to consider one of the most popular paradigms, i.e. *location k-anonymity*.

Figure 2.7 A cloaked region for 10-anonymity.

Given a population of users, location k-anonymity postulates the following requirement, that the user's position disclosed to the service provider must be indistinguishable from the position of at least $k-1$ other users. In practice, the exact user's position must be replaced by a coarser position, normally called *cloaked region*, large enough to contain the position of k-1 other users located nearby at the time the on-line service is requested. Accordingly, the service provider cannot identify the requester of the service based exclusively on the position information. This situation is exemplified in Figure 2.7. For k=10, the position of the single individual is replaced by a larger region (i.e. a cloaked region) containing 10 persons. If the on-line service is requested from this region, the maximum probability of identifying the requester is 1/10. Another prominent feature of this privacy mechanism is that it typically requires a dedicated trusted middleware, the *location anonymizer*, in between the clients and the service provider. The role of the location anonymizer is to collect the position of all the clients, intercept the individual's requests, replace the user's identifier with a pseudo-identifier and finally replace the true position with the dynamically generated cloaked region.

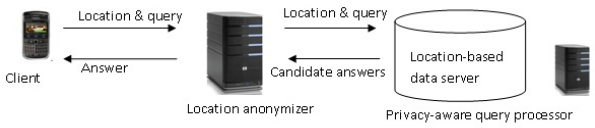


Figure 2.8 The Casper architecture.

One representative solution of this class is the Casper system (Figure 2.8). Casper consists of the location anonymizer and the *privacy-aware query processor* a software component which runs on the server and which resolves user's requests with respect to a position which is not a point as usual, but a region and which returns a set of candidate answers.

A common criticism to location k-anonymity is that it is difficult to gauge which size of k is minimally necessary or sufficient. The higher the value of k, the higher the level of protection but also the loss of position accuracy, i.e. the cloaked region is likely larger. Moreover, the position accuracy varies in time and space based on the distribution of people in space, i.e. if individuals are sparse then the cloaked regions are larger.

Location Privacy Unlike identity privacy, location privacy aims at protecting the position information. The protection strategy is to transmit a position which is somewhat different in the content or in the form from the actual position. In particular, the disclosed position can be fake, cloaked or transmitted using some cryptographic protocol.

- A *fake* position is a position deliberately represented with a wrong value. Privacy is achieved from the fact that the reported position is false. The accuracy and the amount of privacy mainly depend on how far the reported location is from the exact location. For example, the client requesting a service, e.g., "where is the closest restaurant" can transmit to the service provider a fake position in proximity of the actual position and then properly filter out candidate answers.
- An *obfuscated* position (another term for cloaked region) is a coarse region including the exact user's location. Therefore the service provider does know that the user is located in the cloaked region, but has no clue where exactly the user is located. A popular obfuscation method, which is often used in commercial applications, replaces the actual position with a predefined region chosen in a taxonomy of locations at different granularities, e.g., street, zip code area, city. Unfortunately predefined locations can be too broad to ensure an appropriate quality of service, e.g., a zip code region can cover an area of few squared kilometers, or conversely too small to provide privacy guarantees, e.g., a short street. Another simple method obfuscates the position with a circle of user-defined radius and random center containing the actual position. In other solutions, the size of the obfuscated region can be the result of a trade-off between privacy and position accuracy. Moreover the transmission of the position can be also delayed a while to cloak the temporal dimension.

- *Cryptographic protocols* define techniques for the secure collaboration of different parties. An example of cryptographic protocol used for privacy protection in LBS is PIR (Private Information Retrieval). This technique allows users to issue a query without disclosing to the LBS provider the information which is requested as well as the information being returned. In this sense this technique protects both the identity and the location. The method ensures the maximum privacy. However, it incurs high computational costs and can be only applied to certain categories of queries, e.g., the retrieval of stationary objects (i.e. non-mobile objects).

One specific problem that may rise when the position is obfuscated by a coarse region is that consecutive positions in the user's trajectory are correlated, i.e. the presence in one region constrains the position in the subsequent regions. This information can be exploited to prune the obfuscated regions and more precisely delimitate the user's position. To prevent this inference when the maximum speed of the user is known (e.g., the user can be a pedestrian, a car driver, a cyclist and so on) and the movement is frequently sampled, i.e. the position is continuosly reported, an approach is to modify the position in space and time before it is released. This form of privacy leak is also called *velocity-based linkage attack*.

Semantic Location Privacy Semantic location privacy is a form of location privacy which aims at preventing data collectors from identifying the semantic locations in which users stay, e.g., hospitals, religious buildings and so on. Forestalling this type of inference is important for the construction of privacy-aware semantic trajectories.

The motivation behind semantic location privacy is that the sensitivity of positions may vary depending on the nature of places, e.g., the position of a user staying in an oncological clinic is likely *more sensitive* than the position of a user walking along a street. If all the positions are treated as they were sensitive, the protection would be excessive. More effective is to obfuscate only those positions which are perceived as sensitive, disclosed with no change. In this way the loss of position accuracy is limited. This form of obfuscation is called semantic location cloaking. A sound semantic cloaking strategy should guarantee:

- *Semantic diversity*. The user's position cannot be blurred exclusively when the user is inside a sensitive place, but also when he or she is outside. That way, the place in which the user is located remains uncertain. An obfuscated region thus must include places of diverse types.

- *Independence* of the position cloaking method from the user's position. This condition prevents the discovery of the correlation between the cloaked region and the true position, which could be exploited to infer where the user is located.

These guidelines have been embodied in the privacy-preserving framework called Probe (Privacy-aware Obfuscation Environment).

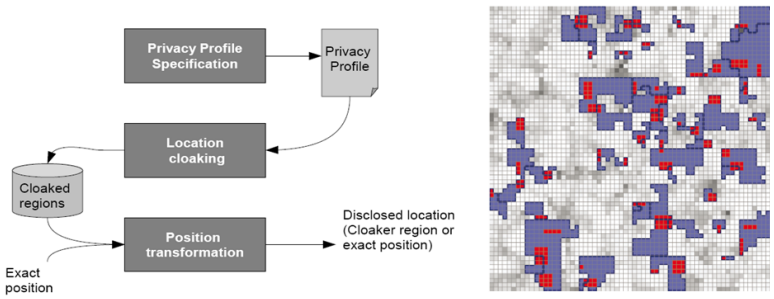


Figure 2.9 The Probe system: (a) the workflow; (b) obfuscated map: The blue polygons represent cloaked regions, the red rectangles sensitive places; the grey background the distribution of population in space.

Figure 2.9 illustrates the workflow of the privacy enforcement process in the Probe system. Users first specify in a privacy profile which categories of points of interest are sensitive (selecting for example from a pre-defined list, e.g., hospitals, religious buildings and so on) along with the degree of privacy desired for each of those categories. For example a privacy degree of 0.1 assigned to hospitals means that the (posterior) probability of locating the user inside a hospital must be less than 0.1. Next, coarse regions are generated satisfying the privacy preferences, independently from the user's position, in order to prevent possible inferences on their reciprocal positions. A sample set of obfuscated regions is shown in Figure 2.9b. Finally, at runtime if the user's position falls inside one of the coarse regions, that region is delivered instead of the exact position. This solution is grounded on a conceptually founded privacy metric. Moreover an additional metric is defined, the utility metric, providing a measure of the spatial accuracy of the cloaked regions. Unlike more traditional obfuscation techniques, the utility measure can be computed prior to any service request. In this way users can tune and balance the amount of privacy with the quality of service.

2.6 Conclusions

In this chapter, we presented techniques for collecting mobility data and handling them appropriately (applying data cleansing, data compression and map matching) so as to produce noise-free and meaningful trajectories (trajectory reconstruction). Finally, privacy issues in mobility data collection and handling were discussed.

We outline next a few research directions that origin from the discussion provided in this chapter.

With respect to *trajectory reconstruction*, future work may include the exploration of intelligent ways to automatically extract proper values of trajectory reconstruction parameters according to a number of characteristics of data sets as well as the extension of this technique so as to be able to identify different movement types (pedestrian, bicycle, motorbike, car, truck etc) and hence to apply customized trajectory reconstruction.

With respect to *privacy issues*, major research directions include: *privacy usability*, i.e. how to provide personalizable, conceptually founded and simple to use privacy mechanisms so to enhance user experience; and *context-aware location privacy*, i.e. tayloring privacy protection based on the context in which individuals are located. While semantic location privacy is a first attempt to introduce the contextual dimension in privacy, this notion can be extended along several directions, for example to account for the temporal and social dimension of privacy.

2.7 Bibliographic Notes

In this section, we distinguish and annotate some works from the literature.

With regard to the data handling approaches, (Yan et al., 2010) proposed a Gaussian kernel-based local regression model to smooth out GPS feeds. (Brakatsoulas et al., 2005) proposed the methodology for map matching that is illustrated in Figure 2.2. (Quddus et al., 2007) proposed a technique for replacing each position of the original trajectory by the point on the network that is the most likely position of the moving object. (Greenfeld, 2002) proposed a method based on topological analysis using the observed position of the individual without assuming any knowledge of the expected traveling route and the speed or heading information supplied by the GPS. Furthermore, (Newson and Krumm, 2009) used Hidden Markov Model approaches to find the most likely road route corresponding to a sequence of positions.

(Meratnia and de By, 2004) proposed the Top-Down Time Ratio (TD-TR)

and Open Window Time Ratio (OPW-TR) algorithms for the compression of spatio-temporal data. (Potamias et al., 2006) proposed the two algorithms, called Thresholds and STTrace, respectively, for online trajectory data compression. (Kellaris et al., 2009) present a different approach by replacing certain episodes of a trajectory by selected shortest paths between the beginning and ending position of these episodes. As for the trajectory reconstruction topic, (Marketos et al., 2008) presented a method for determining different trajectories as part of a trajectory reconstruction manager. On the other hand, (Yan et al., 2011) presented a technique for reconstructing semantic trajectories from the raw GPS mobility records.

With regard to privacy issues, (Gruteser and Grunwald, 2003) introduced the concept of location k-anonymity in the context of LBS; (Jensen et al., 2009) introduced the dichotomy identity privacy vs. location privacy; Casper (Chow et al., 2009) is a major privacy preserving framework supporting location-k anonymity; the velocity-based attack is described in more detail in (Ghinita et al., 2009); (Damiani et al., 2011, 2010) introduces the semantic location cloaking paradigm.