13

Tracking of Moving Objects with Accuracy Guarantees

Alminas Čivilis[1], Christian S. Jensen[2], and Stardas Pakalnis[2]

[1] Vilnius University, Vilnius (Lithuania)
[2] Aalborg University, Aalborg Ost (Denmark)

13.1 Introduction

In step with the increasing availability of an infrastructure for mobile, online location-based services (LBSs) for general consumers, such services are attracting increasing attention in industry and academia [9, 18]. An LBS is a service that provides location-based information to mobile users. A key idea is to provide a service that is dependent on positional information associated with the user, most importantly the user's current location. The services may also be dependent on other factors, such as the personal preferences and interests of the user [3].

Examples of LBSs abound. A service might inform its users about traffic jams and weather situations that are expected to be of relevance to each user. A friend monitor may inform each user about the current whereabouts of nearby friends. Other services may track the positions of emergency vehicles, police cars, security personnel, hazardous materials, or public transport. Recreational services and games, as exemplified by geocaching [8], the Raygun [7] game, may be envisioned. In the latter type of game, individuals catch virtual ghosts (with geographical coordinates) that are displayed on the screens of their mobile phones.

Services such as these rely to varying degrees on the tracking of the geographical positions of moving objects. For example, traffic jams may be identified by monitoring the movements of service users; the users that should receive specific traffic jam or weather information are identified by tracking the users' positions. Some services require only fairly inaccurate tracking, for example, the weather service, while other services require much more accurate tracking, for example, location-based games. In contrast to Chap. 12, this chapter does not consider issues to do with user interface design, but rather concerns support for fundamental functionality that may be exploited by mobile services.

We assume that the users are equipped with wireless devices (e.g. mobile phones) that are online via some form of wireless communication network. We also assume that the GPS [20] positions of the users are available.

To accomplish tracking with a certain accuracy, an approach is used where each wireless device, termed "a moving object," monitors its real position (its GPS

position) and compares this with a local copy of the position that the server-side database assumes. When needed in order to maintain the required accuracy in the database, the object issues an update to the server. The database may predict the future positions of a moving object in different ways.

The challenge is then how to predict the future positions of a moving object so that the number of updates is reduced. This in turn results in reduced communication and server-side update processing. The chapter initially covers three basic techniques for predicting the future positions of a moving object. The first two are point- and vector-based tracking, where an object is assumed to be stationary and to move according to a velocity vector, respectively. In the third approach, segment-based tracking, the future movement of an object is represented by a road segment drawn from a representation of the underlying road network and a fixed speed. A road segment is a polyline, that is, a sequence of connected line segments. So, this representation assumes that a moving object moves along a known road segment at constant speed.

As explained above, a moving object is aware of the server-side representation of its movement. The server uses this representation for predicting the current position of the moving object. The client-side moving object uses the representation for ensuring that the server's predicted position is within the predefined accuracy.

The chapter also covers techniques that aim to improve the basic segment-based approach. The chapter considers modifications of the segments that make up the representation of the road network. The chapter covers the use of anticipated routes for the moving objects, which are represented as (long) polylines, instead of individual segments drawn from the road network representation. The chapter explores the use of acceleration profiles instead of modeling the speed of an object as being constant in between updates.

In summary, the chapter covers three types of techniques that aim to reduce the communication and the update costs associated with the tracking of moving objects with accuracy guarantees, and it reports on empirical evaluations of these techniques and the best existing tracking techniques based on real data.

Chapter 9 concerns access control for LBSs – the reader is referred to that chapter for further information on this highly relevant aspect of tracking.

The coverage of tracking techniques is primarily based on proposals by Čivilis et al. [4, 5] and Jensen et al. [11]. These works share the general approach with Wolfson et al. [22, 24]. The chapter offers results of new empirical performances studies, based on two real GPS data sets, of the techniques presented. A more detailed coverage of related studies is given in Sect. 13.8.

The presentation is organized as follows. Section 13.2 describes the general approach to tracking and describes the data sets used in experiments. Section 13.3 describes point-, vector-, and segment-based tracking. Section 13.4 covers improvements in the segment-based approach using road network modifications. Sections 13.5 and 13.6 present techniques for update reduction using routes and acceleration profiles, respectively. Section 13.7 is a summary, and Sect. 13.8 covers commercial developments and points to further readings.

13.2 Background

We first describe the general approach to tracking that we use. Then we describe the real-world GPS and road network data that we use for evaluating the different tracking techniques.

13.2.1 Tracking Approach

We assume that moving objects are constrained by a road network and that they are capable of obtaining their positions from an associated GPS receiver. Moving objects, also termed "clients," send their location information to a central database, also termed "the server," via a wireless communication network. We assume that disconnects between client and server are dealt with by other mechanism in the network than the tracking techniques we consider. When a disconnect occurs, these mechanisms notify the server that may then take appropriate action.

After each update from a moving object, the database informs the moving object of the representation, or prediction function, it will use for the object's position. The moving object is then always aware of where the server thinks it is located. The moving object issues an update when the predicted position deviates by some threshold from the real position obtained from the GPS receiver. We term this the "shared prediction-based approach" to tracking.

Figure 13.1 presents a UML activity diagram for this tracking approach (activity diagrams model activities that change object states). The object initially obtains its location information from its GPS receiver. It then establishes a connection with the server and issues an update, sending its GPS information and unique identifier to the server.
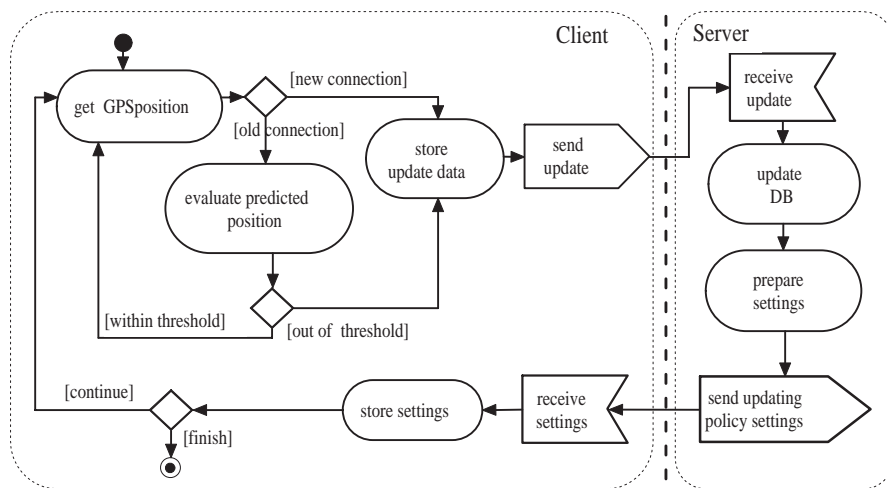

Fig. 13.1. Tracking scenario diagram

Having received this update, the server determines which tracking technique and threshold to be used for the object (these are predefined), and it stores the information received from the object in the database. If segment-based tracking is to be used, the server also uses map matching to determine on which road segment the object is moving. The server then sends its representation, or prediction, of the object's current and future position to the object.

Having received this information from the server, the object again obtains its actual, current location information from the GPS receiver. It then calculates its predicted position using the representation received from the server, and it compares this to the GPS position. If the difference between these two exceeds the given threshold, the client issues an update to the server. If not, a new comparison is made. This procedure continues until it is terminated by the object. Although the server may also initiate and terminate the tracking, we assume for simplicity that the object is in control. This aspect has no impact on the chapter's exposition.

13.2.2 Data Description

As mentioned, we assume that GPS is used for positioning of the moving objects. In making this assumption, we note that Galileo-based positioning [19] and hybrid GPS/Galileo-based positioning are likely to work even better when they become available. In experiments, the results of which will be reported upon throughout the chapter, we used two data sets of GPS logs. Both were obtained by installing GPS receivers together with small computers in a number of vehicles. The positions of the vehicles were recorded every second while the vehicles were driving. Positions were not recorded for a vehicle when its engine was turned off.

The first GPS data set stems from a Danish intelligent speed adaptation project called "INFATI" [10]. A total of 20 GPS equipped cars were participating in the project, and their positions were recorded during a period of approximately 8 weeks. Cars were driving in Aalborg, Denmark area, an area with a population of about 140,000 inhabitants. This data set represents the behavior of vehicles traveling in semiurban surroundings. Here, the average trip length is 9.5 km (continuous driving ignoring pauses shorter than 5 minutes is considered to be one trip). The part of the INFATI data set used in the experiments reported in this chapter consists of about 500,000 GPS records, and the total trip length is about 9,000 km.

The second GPS data set stems from a road-pricing project called "AKTA" [16]. Here, the participating cars were driving in the Copenhagen, Denmark area. This data set represents the behavior of vehicles traveling in a larger urban area. Here, the average trip length is 17.9 km, and the part of the data set used consists of about 4,000,000 GPS records, corresponding to a total trip length of about 67,000 km.

For the experiments, we also used digital road networks obtained from both projects. The initial road networks were composed of sets of segments, where each segment corresponds to some part of a road between two consecutive intersections or and intersection and a dead end. A segment consists of a sequence of coordinates, that is, it is a polyline. Further, the road networks are partitioned into named roads or

streets, meaning that each segment belongs to precisely one road or street. Each segment identifies its road or street by means of a street code. Chap. 2 offers additional detail on more comprehensive modeling of road networks.

13.3 Fundamental Tracking Techniques

We proceed to describe three tracking techniques that follow the scenario described in Sect. 13.2.1 but differ in how they predict the future positions of a moving object. These were covered by Čivilis et al. [4]; minor variations of the first and third of these were also studied by Wolfson and Yin [24] (see Sect. 13.8 for additional discussion).

13.3.1 Point-based Tracking

Using this technique, the server represents an object's future position as the most recently reported position. An update is issued by an object when its distance to the previously reported position deviates from its current GPS position by the specified threshold. Thus, the movement of an object is represented as a "jumping point." This technique is the most primitive among the techniques presented, but it may well be suitable for movement that is erratic, or undirected, with respect to the threshold used. An example is the tracking with a threshold of 200 m of children who are playing soccer.

The algorithm for point-based tracking, **PP** (Predict with Point), is simple.

Algorithm 13.3.1 PP(mo)

(1)   **return** $mo.p$

As the prediction is constant, the predicted position is the same as the input position. Here $mo.p$ is the position of the moving object.

13.3.2 Vector-based Tracking

In vector-based tracking, the future positions of a moving object are given by a linear function of time, that is, by a start position and a velocity vector. Point-based tracking then corresponds to the special case where the velocity vector is the zero vector.

A GPS receiver computes both the speed and the heading for the object it is associated with — the velocity vector used in this representation is computed from these two. Using this technique, the movement of an object is represented as a "jumping vector." Vector-based tracking may be useful for the tracking of "directed" movement.

Algorithm **PV** (Predict with Vector) predicts the location of the given object mo at a given time t_{cur}.

Algorithm 13.3.2 PV(mo, t_{cur})

(1) $p_{pred} \leftarrow mo.p + mo.\overline{v}(t_{\text{cur}} - mo.t)$
(2)   **return** p_{pred}

The result of the algorithm is the location of *mo* at time t_{cur}. The predicted location is calculated by adding the time-dependent traveled distance ($t_{\text{cur}} - mo.t$) to the starting point in the direction of vector $\overline{v}$.

13.3.3 Segment-based Tracking

Here, the main idea is to utilize knowledge of the road network in which the objects move. A digital representation of the road network is thus required to be available. The server uses the GPS location information it receives from an object to locate the object within the road network. This is done by means of map matching, which is a technique that positions an object on a road network segment, specified as a distance from the start of that segment, based on location information from a GPS receiver.

In segment-based tracking, the future positions of an object are given by a movement at constant speed along the identified segment that is represented as a polyline. The speed used is the speed most recently reported by the client. When or if a predicted position reaches the end of its segment, the predicted position remains there from then on. In effect, the segment-based tracking switches to point-based tracking.

Special steps are needed to ensure robustness when segment-based tracking is used. In particular, if for some reason, a matching road segment cannot be found when a moving object issues an update, the segment-based approach switches temporarily to the vector-based approach that is always applicable. On the next update, the server will again try to find a matching road segment in the database. This arrangement ensures that segment-based tracking works even when map matching fails. Map matching may fail to identify a segment for several reasons. For example, the map available may be inaccurate, or it may not cover the area in which the client is located.

Using segment-based tracking, the movement of an object is represented as a set of road segments with positions on them, and as jumping vectors in case map matching fails. This technique takes into account the shape of the road on which an object is moving – an object thus moves according to the shape of the road.

Algorithm **PS** (Predict with Segment) is defined as follows.

Algorithm 13.3.3 PS(mop, t_{cur})

(1) $m_{\text{pred}} \leftarrow mop.m + mop.plspdt_{\text{cur}} - mop.t)$
(2)   **if** $m_{\text{pred}} >= \mathcal{M}mop.pl, p_n)$ **then return** $mop.pl.p_n$
(3)   **elsif** $m_{\text{pred}} <= 0$ **then return** $mop.pl.p_0$
(4)   **else**
(5)   **return** $\mathcal{M}^{-1}(mop.pl, m_{\text{pred}})$
(6)   **end if**

Here, an object's position is given by a polyline *mop.pl* and a measure *mop.m*. The predicted location is given as a new measure that is equal to the old measure (line 1) plus the distance traveled since the last update, $t_{cur} - mop.t$. The traveled distance is negative if the object moves against the direction of the polyline. If the new measure is outside the polyline (lines 2 and 3), the result is one of the end points of the polyline.In this way, the position prediction stops at a boundary point of the polyline (the first point $pl.p_0$ or the last point $pl.p_n$). Function $\mathcal{M}$ calculates the measure value on a given polyline of a given coordinate point. Otherwise, the coordinate point of m_{pred} is calculated with the inverse function $\mathcal{M}^{-1}$ that calculates the coordinate point of a given measure value on a given polyline.

13.3.4 Comparison of Tracking Techniques

Results of experimental studies with the three tracking techniques are presented in Fig. 13.2. Here both the INFATI and AKTA data sets, described in Sect. 13.2.2, are used. For the experiment, the INFATI data set contributed with approximately 500,000 GPS positions collected from five cars. The AKTA data set contributed with 4,000,000 GPS positions obtained from five cars. Labels for results obtained using the INFATI data start with 'i' while labels for results obtained using the AKTA data start with an 'a.'
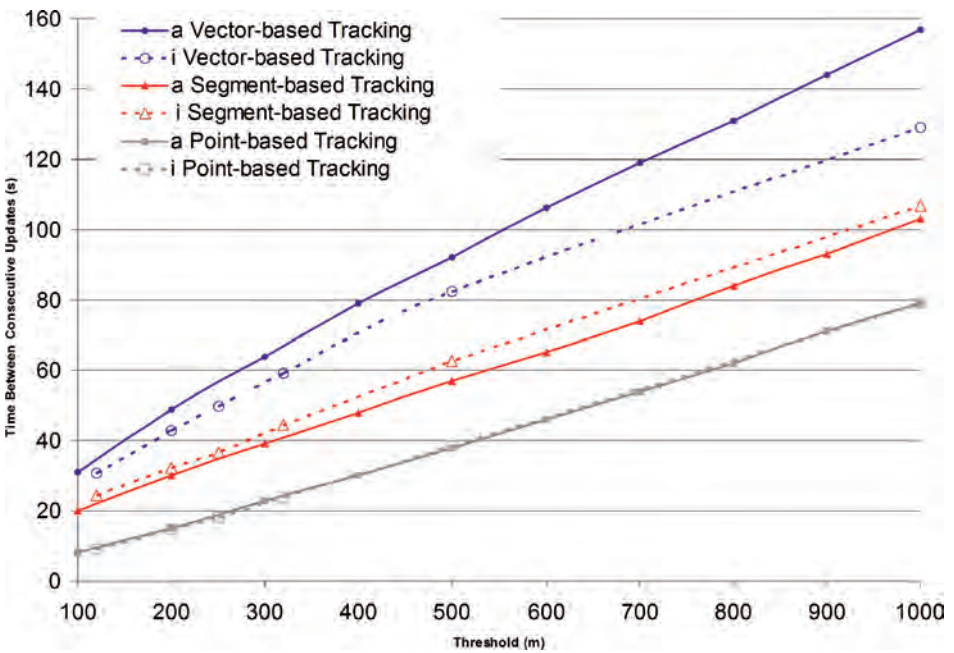

Fig. 13.2. Comparison of tracking techniques

The results were obtained by simulating the scenario described in Sect. 13.2.1 with thresholds ranging from 100 to 1,000 m. Specifically, the movement of each car was simulated using the log of GPS positions for the car. So a client program and a server program interact, and a simple experiment management system is in charge of the bookkeeping needed to obtain the performance results. Instead of obtaining GPS positions from a GPS device in real time, the client program utilizes the GPS logs, which of course makes the simulation much faster than the reality being simulated. The bookkeeping involves the counting of updates sent from the client program to the server program and keeping track of time.

All performance studies reported in this chapter follow this pattern. The studies differ in the specific GPS data and road networks used, and in the tracking policies used.

In Fig. 13.2, accuracy threshold values in meters are on the *x*-axis. The client obtains a GPS position from the GPS device every second and performs a comparison between the GPS position and the predicted position. The *y*-axis then gives the average number of seconds between consecutive updates sent from the client to the server to maintain the required accuracy.

It is seen that the time between updates increases as the accuracy threshold increases, that is, as the required accuracy decreases. Point-based tracking shows the worst performance. The largest improvement of the segment- and vector-based techniques over the point-based technique is for smaller thresholds, while for larger thresholds the improvement is smaller. For thresholds below 200 m, segment- and vector-based tracking policies are more than two times better than point-based tracking.

Segment-based tracking is outperformed because the road segments in the underlying road network are relatively short, having an average length of 174 m. For example, this means that a relatively straight road is represented by several segments. In this case, vector-based tracking may need less updates. So, although vector-based tracking is simpler and performs slightly better, we find it likely that it is possible to improve segment-based tracking to become the best.

In addition, segment-based tracking, by relating the location of a moving object to the underlying road network, offers other advantages that are as follows:

- Buildings, parking places, traffic jams, points of interest, traffic signs, and other road-related information that is mapped to the road network can easily be associated with the location of a moving object.
- Road network-based distances can be used in place of Euclidean distances.
- Acceleration profiles, driver behavior on crossroads, and other road-related data that increase the knowledge about the future positions of moving objects can be exploited.

Consequently, a promising direction for obtaining improved tracking is to continue in the direction of segment-based tracking.

13.4 Modifying the Road Network Representation

Recall that with segment-based tracking, the predicted position of an object moves at constant speed along a segment drawn from the underlying representation of the road network until it reaches the end of the segment, at which time the predicted position remains at the end of the segment. The experimental study reported in Sect. 13.3 indicates that the numbers of updates in segment-based tracking are closely correlated with the numbers of segment changes. This motivates attempts to modify the underlying road network representation so that less segment changes occur.

We proceed to cover several modifications of the road network representation. The main idea is to connect road segments in such a way that moving objects would have to change segments as few times as possible as they travel in the road network. We first describe three modification approaches and then report on experimental studies with these approaches.

13.4.1 Modification Approaches

The general idea of the road network modification is to iterate through all segments in the road network to be modified according to some specified ordering. During each iteration, the modification algorithm orders all available segments and then tries to extend the topmost, or current, segment with other segments. To do this, the algorithm identifies all the existing segments that start or end at the start or end of the current segment and extends the current segment with the most attractive of such segment(s) according to some other specified ordering. A current segment that has been extended is considered in the next iteration, but the segment(s) that were used for the extension are disregarded. A current segment that has not been extended becomes part of the final result and is not considered any further. Several different ordering strategies can be considered. Details on the modification approaches beyond what is reported next are given elsewhere [5].

Street code-based Approach

As described at the end of Sect. 13.2.2, a named street (e.g. "Main Street") is represented by many segments, and the segments have street codes that identify the named street that they represent part of.

The ordering in the street code-based (SSC) approach exploits the availability of street codes for the segments. The idea is to give priority to connecting polylines with the same street code. In this way, longer road segments are constructed that tend to correspond to parts of the same street. In cases where there are several candidates with the same street code, priority is given to the shortest polyline. This strategy reduces the probability that unconnected polylines will be short.

Tail Disconnection Approach

The SSC does not distinguish between main roads and side streets. The observation that motivates the tail disconnection approach (TSC) is that moving objects can

be assumed to be moving on main roads most of the time. This approach thus first connects polylines while disregarding side streets, termed "tails," and it only subsequently takes the tails into account.

To be more precise, a polyline *pl* in a set of polylines *rn* is a *tail* if at least one delimiting point of *pl* is not connected to any delimiting point of any other polyline in *rn*. Tails are also termed "first-level tails." The *i*th level tails in *rn* are those polylines that are tails in the set obtained by subtracting all tails at lower levels than *i* from *rn*.

A few observations are in order. If a road network has a purely hierarchical structure, each polyline is a tail at some level. Polylines that belong to a circular structure in a road network, that is, a structure where each constituent polyline is connected at both ends, are not tails. A highest tail level is assigned to all non-tail polylines. The ranking is based on the tail level.

Direction-based Approach

The last approach takes into account the directions of the candidate polylines at the connection points. The idea is that moving objects are expected to prefer to be moving as directly as possible toward their destinations, which means that they will tend to move as straight as possible and by making as few turns as possible.

This approach thus gives preference to polylines that continue in the same direction as much as possible when extending a polyline. Put differently, preference is given to polylines with a direction at a connection point that has as small an *angle* as possible with respect to the direction of the polyline to be extended, again at the connection point.

Figure 13.3 explains this further. The property *angle* denotes the angle between the polyline being extended and a candidate for use in the extension. The figure contains two such candidates and thus has two angles. Specifically, the line segment at the connection point of the polyline to be extended is itself extended toward the candidate polyline. This extension corresponds to a straight extension of the polyline's line segment at the connection point. The property *angle* is then the angle between the extended line segment and the line segment of the candidate polyline at the connection point. A small angle is thus preferable.
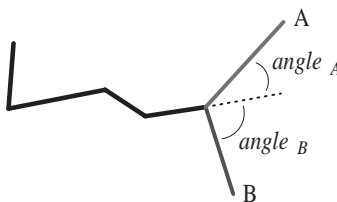

Fig. 13.3. The *angle* property between polylines

13.4.2 Comparison of Approaches

The goal of the road modification approaches described in the previous section is to connect the polylines of road segments into longer polylines, so that moving objects travel on fewer polylines. In doing this, we assume that objects in a road network move mostly along the main roads. We proceed to evaluate the results of the road network modifications in terms of how well the constructed polylines correspond to the main roads.

All policies succeeded in connecting short polylines into longer ones. The polylines created by the SSC were the worst in connecting the main roads. In residential and other areas, it is common for a main road and its side streets to have the same street code. (Or put differently, a road may have many side roads.)

Figure 13.4 offers a comparison of the update performance for segment-based tracking using the unmodified road network and the road networks resulting from the
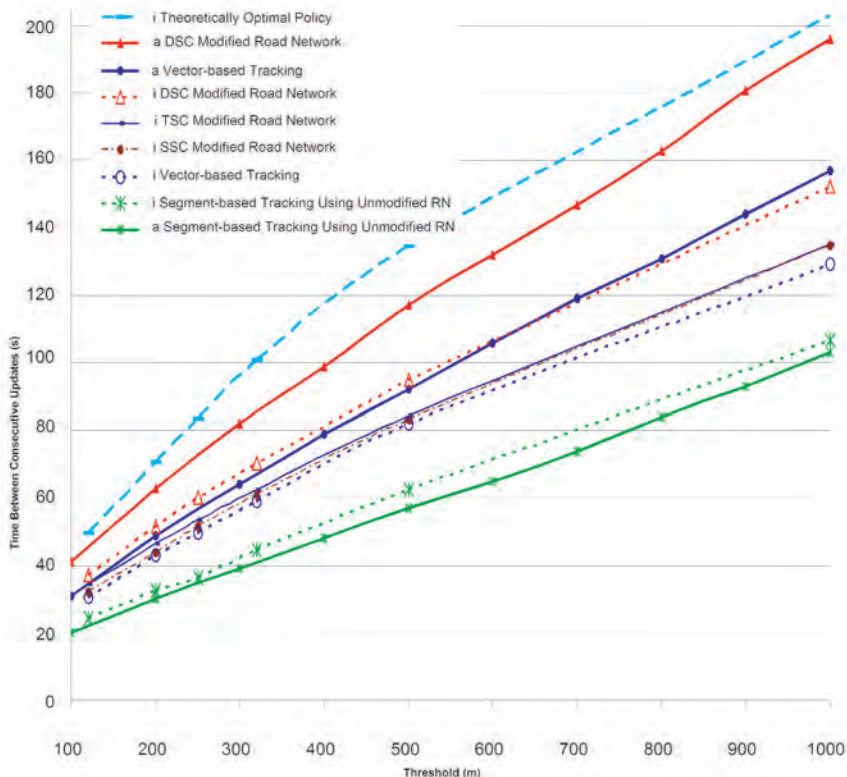

Fig. 13.4. Comparison of road network modification approaches

application of SSC, TSC, and DSC. Vector-based tracking is also included. To avoid clutter, only the most interesting techniques are illustrated for the AKTA data.

In the comparison, 568,307 GPS records were used from the INFATI data set and about 4,000,000 GPS records from the AKTA data set. The curves show experimental results using thresholds ranging from 100 to 1,000 m.

All three road network modifications increase the performance of segment-based tracking, which then outperforms vector-based tracking. Segment-based tracking has the best performance when using the road network resulting from the direction-based modification.

The performance of a theoretical form of tracking that is optimal under the assumption that the speed of an object is modeled as being constant between updates is also included in Fig. 13.4. This technique is explained in Sect. 13.5.

13.5 Update Reduction Using Routes

The focus of this section is on the use of the routes of moving objects for update reduction. We first describe the theoretical, constant-speed optimal form of tracking mentioned in the previous section. Then we consider the use of an object's routes, which are 'long' segments, during segment-based tracking instead of the previous use of road-network segments.

13.5.1 Theoretical, Constant-speed Optimal Tracking

One may distinguish between updates based on the outcomes of the associated map matching. Recall that in segment-based tracking, when the server receives an update at a position p_i, it attempts to map match the position onto the road network rn to find the most probable polyline mpl and point mp on it.

Let $\mathbf{MM}$ be the map matching function and $\mathbf{MM}(p_i, rn) = (mpl, mp)$. If $\mathbf{MM}$ fails to identify a polyline and point, tracking is done in vector mode. Assuming that the map matching is successful and $(\mathbf{MM}(p_{i-1}, rn)).mpl = (\mathbf{MM}(p_i, rn)).mpl$, where position p_{i-1} is that of the previous update, we say that the update is caused by *speed*. If the polylines differ, we say that the update is caused by a *segment change*.

The theoretical, constant-speed optimal tracking introduced here indicates how it is possible to achieve few updates with segment-based tracking in the best case, which occurs when a moving object travels on only one segment and no updates occur due to segment changes. The technique is optimal under the assumption that the speed of an object is modeled as being constant between updates.

This technique is interesting because it offers a measure of optimality. However, the technique is useful for comparison purposes only and it is not a practical technique. The technique is impractical because it assumes that the entire polyline along which a vehicle will ever move is known in advance. We are able to use this technique here because we have the entire GPS logs for each vehicle. Using these, we simply construct (very long) polylines that precisely track each vehicle "ahead of time." In practice, GPS positions are received in real time.

In Fig. 13.4, the curve for the constant-speed optimal policy gives the lower bound for the number of updates needed by the segment-based policy. The deviation of the segment-based policy using the non-modified road network from the optimal case is substantial. Using the modified road networks, the performance is significantly closer to the optimal case. For example, for a threshold of 200 m, the use of the road network modified using the direction-based approach increases the average time duration in-between consecutive updates from 32 vs. 30 s to 52 vs. 63 s (for the INFATI vs. AKTA data sets) in comparison to the use of the unmodified road network.

13.5.2 Practical Tracking Using Routes

We may assume that individuals who travel do so to reach a destination; and the same routes are often used multiple times. For example, a person going from home to work may be expected to frequently use the same route. This behavior is confirmed by the GPS logs available [10, 16].

Taking advantage of knowledge of the routes used by a moving object holds potential for reducing the number of updates caused by segment changes. Since a route is a sequence of connected, partial road segments, a route is represented simply as a polyline. Therefore, segment-based tracking that is applicable to any polylines is also directly applicable to routes.

All that is needed is to collect the routes of each user. Routes may be obtained via a navigation system. This case may occur when the user travels in unfamiliar surroundings. When the user travels in familiar surroundings, it is likely that the user travels along a route that was used previously. In this case, a system that gathers the user's routes and associates usage meta-data (e.g. times of the day and days when the routes are being used) with these can be used [2]. Such a system is able to return likely routes on request.

When using segment-based tracking with routes, we effectively assume that we know the future positions of an object. This is like in the theoretical, constant-speed optimal policy. The differences are that the polylines that represent routes are created from the road network, not from GPS logs, and that deviations from the assumed route are handled. Specifically, if an object deviates from its route, this is treated simply as a segment change. This will then most likely trigger an update.

When the route of an object is guessed successfully, the theoretical technique and the practical, segment-based technique have essentially the same performance. Slight deviations may occur because the routes used by the techniques differ: the routes used by the theoretical technique are constructed from GPS points and are more detailed, and slightly longer than those used by the segment-based technique.

Figure 13.10 illustrates the performance of segment-based tracking using routes for the INFATI and AKTA data. We conclude that exploiting knowledge of the routes used by an object can eliminate virtually all updates caused by segment changes that significantly improve the performance of the segment-based policy.

## 13.6 Update Reduction Using Acceleration Profiles

Even if the future trajectory of an object is known precisely and updates caused by segment changes thus are eliminated, updates still occur due to variations in speed. The reason is that segment-based tracking assumes that objects move at constant speed – it takes an update to change the speed.

In this scenario, the modeled speed of an object moving along a road is a stair function. Figure 13.5 presents the variation of a car's speed along a part of its route from home to work. The stepwise constant speed is the one used by the segment-based policy with a 90 m threshold. Each new step in the stair function represents an update. The density of the steps depends on the threshold – smaller thresholds yield more updates.

It is reasonable to expect that more accurate modeling of the speed variation of an object along its route, for example, using averages of the speeds during past traversals of the route, can help better predict the future position of the object as it moves along the route. Figure 13.6 illustrates the speed variation of one car as it traverses part of its route from home to work (the same car as in Fig. 13.5). Here, the thin lines represent the speeds for 20 traversals of the route, and the solid line represents the average speed along the route.

The figure reveals a clear pattern of how fast the car drives along different parts of the route. The geometry of the route, the driver's habits, and the traffic situation are probably the primary causes for the observed behavior. Figure 13.7 displays the geometry of the partial route. The figure contains distance measures that allow the reader to correlate the geometry with the patterns displayed in Fig. 13.6.

The first deceleration of the car happens in the preparation for negotiating a rotary. Then the car accelerates, decelerates, makes a left turn, enters a highway, and
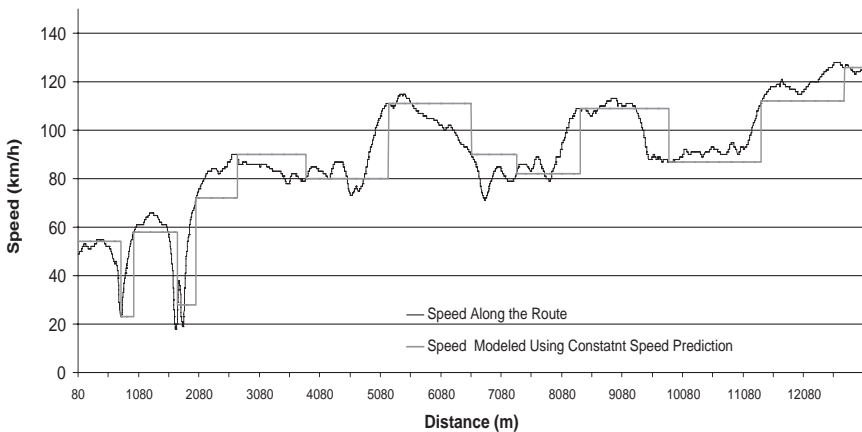
Fig. 13.5. Speed modeling using constant speed prediction
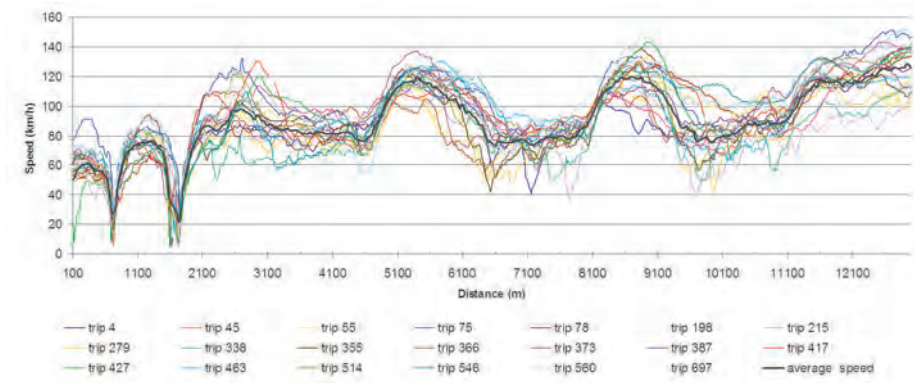
Fig. 13.6. Speed pattern for 20 traversals of a partial route

accelerates further. Note that even on the highway, the car's speed is influenced by exits from the highway and that a clear pattern can be seen. We expect this type of behavior to be typical.

The clear pattern in Fig. 13.6 indicates that tracking with better performance can be achieved by more accurate modeling of the predicted, future speed of a moving object.

Figure 13.8 illustrates another part of the same route with the same 20 traversals where no clear speed pattern exists and where constant speed prediction may work well, or at least better than prediction using variable speeds.

We consequently create an acceleration profile for capturing the average speed variation of the movement of an object along a route. While we create a profile for each combination of a route and an object using the route, it is also possible to assign profiles to the road network that are to be applied to all moving objects and for all uses of the segments of the road network. Such profiles should then be time varying. A separate software component is assumed to be present that generates frequently used routes for the moving objects being tracked [2]. Having this as a separate component is reasonable, as routes are useful for other tasks than tracking.

An acceleration profile consists of acceleration values together with the distance intervals during which these values apply. A profile is created by first dividing the average speed variation along a route into intervals where the acceleration changes sign (i.e. from positive to negative or vice versa). Then the average acceleration is calculated for each interval. An acceleration profile *apf* is then a sequence of $n + 1$ measures m_i and n accelerations a_i, $(m_0, a_0, \ldots, m_{n-1}, a_{n-1}, m_n)$. Acceleration a_i is valid at interval (m_i, m_{i+1}).

To see how an acceleration profile is used, assume that an object moves with speed v_0 and that its current location (measure) along the route is m_0 distance units after the start of the route, where m_0 belongs to the interval $[m_{begin}, m_{end})$ in which the acceleration profile has acceleration value a. Then the predicted position m_{pred}
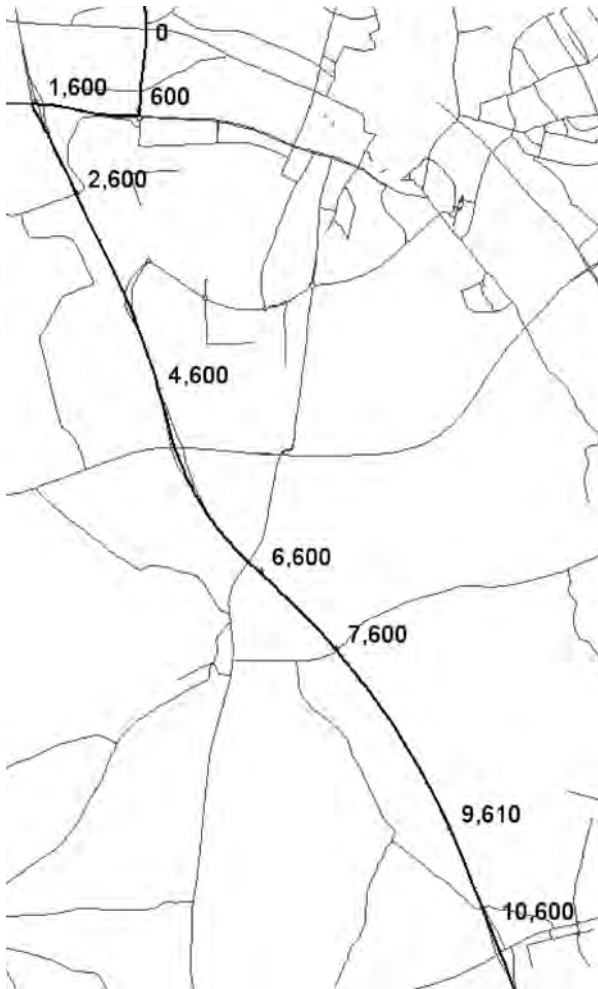
Fig. 13.7. Geometry of partial route

and the speed v_{pred} of the object within the interval $[m_{begin}, m_{end})$ at time t is given by: $m_{pred} = m_0 + v_0 t + (a/2)t^2$ and $v_{pred} = v_0 + at$.

Figure 13.9 exemplifies speed modeling when using an acceleration profile. The figure concerns the movement of one moving object along a route. We assume that segment-based tracking with a 90 m threshold is used. In this figure, the light vertical dotted lines mark updates. To provide better insight into the behavior of the policy used, we include the deviation between the real position of the moving object and its position as predicted by the policy.

The algorithm 'Predict Positions with Segments and Accelerations' (**PPSA**) takes two parameters as input, *mopa* and *t*, where the first parameter is a structure with five elements that are as follows: (1) A polyline, *mopa.pl*, that specifies the
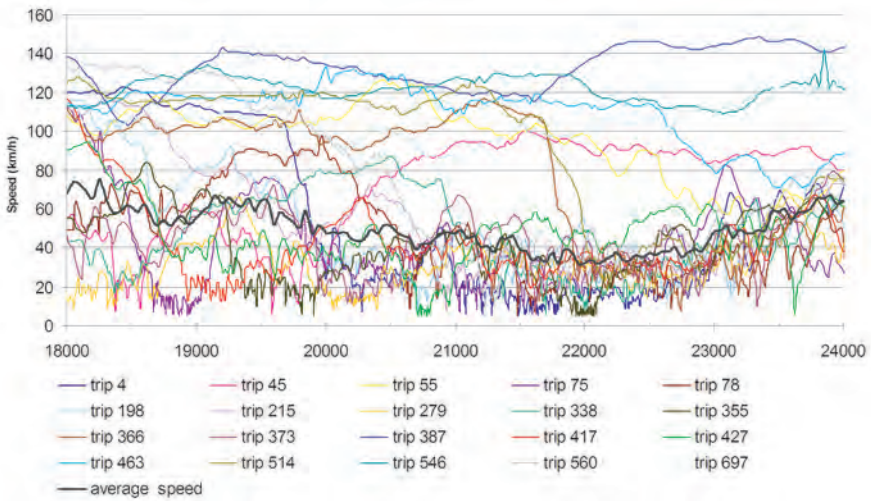
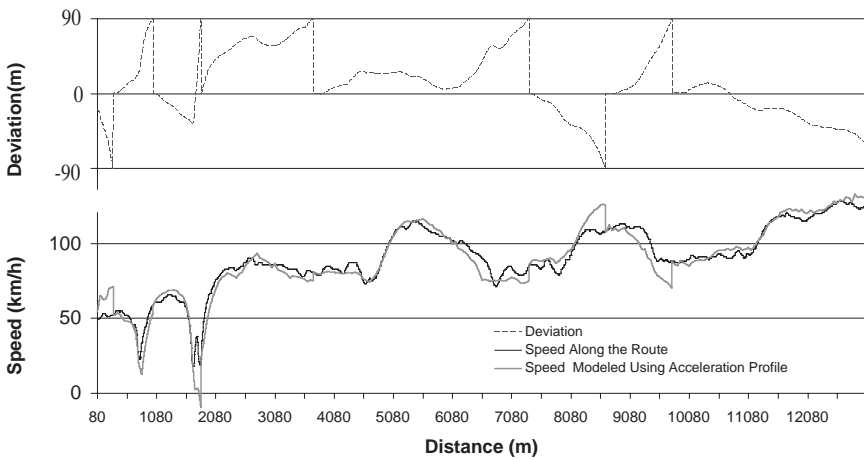**Fig. 13.8.** Speed pattern for 20 traversals of a partial route

Fig. 13.9. Speed modeling using acceleration profile

geometrical representation of the moving object's route, (2) an acceleration profile, *mopa.apf*, for speed prediction along the route, (3) the location of the client, *mopa.m*, given as a measure value on the route, (4) the speed, *mopa.plspd*, of the object, and (5) the time, *mopa.t*, when the location and speed are acquired. Parameter $t > mopa.t$ is the time point for which the location of the object should be calculated. The result is the coordinates of predicted location of the object at time t.

Algorithm PPSA(*mopa*, *t*)

1. $m_{pred} \leftarrow mopa.m$
2. $v_{pred} \leftarrow mopa.plspd$
3. $t_{pred} \leftarrow t - mopa.t$
4. **while** $t_{pred} > 0$ **do**
5. $accel \leftarrow$ **getAcceleration** $(m_{pred}, mopa.apf)$
6. $S \leftarrow accel.end - m_{pred}$
7. $dt \leftarrow 0$
8.   **if** $v_{pred}^2 + 2 \cdot accel.a \cdot S \geq 0 \wedge accel.a \neq 0$ **then**
9. $dt_1 \leftarrow (-v_{pred} + \sqrt{v_{pred}^2 + 2 \cdot accel.a \cdot S})/accel.a$
10. $dt_2 \leftarrow (-v_{pred} - \sqrt{v_{pred}^2 + 2 \cdot accel.a \cdot S})/accel.a$
11. $dt \leftarrow \max(\{0, \min(\{dt | dt \in \{dt_1, dt_2\} \wedge dt > 0\})\})$
12.   **if** $dt = 0$ **then** $dt \leftarrow S/v_{pred}$
13. $accel.a \leftarrow 0$
14.   **if** $t_{pred} < dt$ **then** $dt \leftarrow t_{pred}$
15. $m_{pred} \leftarrow m_{pred} + v_{pred} \cdot dt + accel.a \cdot dt^2/2$
16. $v_{pred} \leftarrow v_{pred} + accel.a \cdot dt$
17. $t_{pred} \leftarrow t_{pred} - dt$
18. **if** $m_{pred} \geq \mathcal{M}mopa.pl, mopa.pl.p_{end})$ **then return** $mopa.pl.p_{end}$
19. **return** $\mathcal{M}^{-1}(mopa.pl, m_{pred})$

The algorithm first initializes temporary variables. Variables m_{pred} and v_{pred} are set to contain starting location and speed of the moving object, and variable t_{pred} initially holds the time elapsed since the time when the moving object's location was acquired. The object's movement should be predicted for this duration of time. In general, several acceleration intervals are traversed during this duration of time, meaning that different acceleration values should be applied during the prediction. The algorithm iteratively calculates the time duration required to pass through each acceleration interval and reduces prediction time t_{pred} with this duration. When the prediction time duration is exhausted (line 4), the loop stops, and the algorithm calculates and returns the coordinates of the predicted location.

In line 5, acceleration value a for the predicted location of the object m_{pred} and boundary point *end* of the acceleration interval where acceleration value a applies are retrieved and stored in *accel*; these are returned by function **getAcceleration**. In the case where m_{pred} is equal to boundary point m_i, the boundary point m_{i+1} of the next acceleration interval is returned. If there are no more acceleration intervals, an acceleration value of 0 is returned, and the boundary point is set to ∞. Note that m_{pred} is initially equal to the location of the object at the time of the update (line 1).

In line 6, the distance S to the end of the acceleration interval with acceleration *accel.a* is calculated.

The time dt required for the object to reach the end of the acceleration interval (moving with acceleration *accel.a*) is calculated in lines 9–11. This time is calculated using the quadratic equation $accel.a \cdot dt^2/2 + v_{pred} \cdot dt - S = 0$. It has solutions only if $v_{pred}^2 + 2 \cdot accel.a \cdot S \geq 0$ (line 8), and only positive solutions are valid, as the

meaning of the solution is time. If there are two positive solutions, the solution with the smaller value is the valid one (line 11). If the equation has no valid solution, the result *dt* is equal to 0. In this case, prediction using constant speed is performed (lines 12 and 13).

After the time required to reach the end of the acceleration interval is calculated, this time is compared to the remaining prediction time t_{pred}. If the time left for which prediction should be done, t_{pred}, is less than the time required to go a distance S, the algorithm does prediction only for time t_{pred} (line 14). Lines 15 and 16 then calculate the predicted location m_{pred} and the speed v_{pred}. The prediction time is reduced in line 17, and the loop is repeated if $t_{pred} > 0$.

Finally, the coordinates corresponding to location m_{pred} are calculated and returned. This is done in lines 18 and 19. If the predicted location m_{pred} is beyond the end of the route as described by polyline *mopa.pl* (line 18), the end point of the polyline is returned. This is done by comparing the predicted measure on the polyline with the measure of the end point $pl.p_{end}$ of the polyline.

Experimental results for the segment-based policy using routes and acceleration profiles are presented in Fig. 13.10. These experiments are based on approximately 57,000 GPS records from the INFATI data set that correspond to the movement of five cars along different routes. In addition, approximately 36,000 GPS records from the AKTA data set, corresponding to one car moving along one route, were used. The experiments show that the use of acceleration profiles is able to improve the performance. This illustrates that when an object's movement has a clear acceleration profile and this profile is known, it is possible to more accurately predict the positions
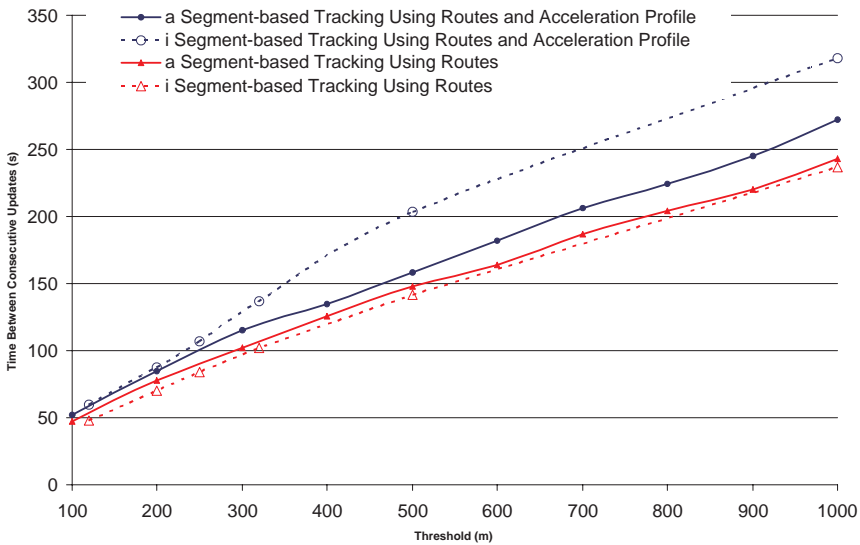

Fig. 13.10. Results using routes with and without acceleration profiles

of the object. For example, using a threshold of 500 m, the average time between up-
dates is increased from 141 to 203 s with the INFATI data and from 148 to 158 s with
the AKTA data. The lower benefit from using acceleration profiles for the AKTA data
is likely to be due to congestion (see Fig. 13.8) as well as the majority of routes being
on the highway where speed patterns are not so clear.

We note that with acceleration profiles, we outperform the previously introduced
theoretical technique that is optimal only under the assumption of constant-speed
prediction.

In closing, it is also worth considering a few alternatives for the speed model-
ing and some implications of the alternative presented. In reality, the travel speed
associated with a road segment varies during the day and different drivers may well
negotiate the same segment with different speeds. By associating acceleration pro-
files with routes that are specific to individual drivers, we capture the variation among
drivers. And because the same route (e.g. from home to work or from work to home)
is typically used during the same time of the day, the variation of speeds during the
day is also taken into account fairly well. Next, if significant variations exist within
the observations based on which the acceleration profile of a route is constructed,
it is possible to create several speed profiles, for example, so that rush-hour and
non-rush-hour profiles are available.

13.7 Conclusions

This chapter presents and empirically evaluates a range of techniques for the tracking
of moving objects, including point-, vector-, and basic segment-based tracking. The
proposed techniques are robust and generally applicable; they function even if no
underlying road network is available or if map matching is not unsuccessful, and
they apply to mobile objects with even stringent memory restrictions.

The performance of basic segment-based tracking is sensitive to the segmentation
of the road network representation used and to the speed variations of the moving ob-
jects. Based on these observations, the chapter describes several techniques that aim
to reduce the number of updates needed for segment-based tracking with accuracy
guarantees. They are the following:

- *Road Network Modification*. The segment-based representation of the underly-
  ing road network used in segment-based tracking is modified with the goal of
  arriving at a segmentation that enables objects to use as few segments as possi-
  ble as they move in the road network. This then reduces the number of updates
  caused by segment changes.
- *Use of Routes*. A route is a polyline constructed from (partial) road network seg-
  ments that capture an object's entire movement from a source to a destination.
  As segments are themselves polylines, segment-based tracking readily accom-
  modates the use of routes. Routes are specific to individual moving objects, and
  the use of routes is expected to reduce the number of updates caused by segment
  changes.

- *Use of Acceleration Profiles.* An acceleration profile divides a route into intervals with constant acceleration and thus enables quite accurate modeling of the speed of an object as it travels along a route. The idea underlying the use of acceleration profiles is to reduce the number of updates incurred by speed variations.

Experimental performance studies using real GPS logs and corresponding real road networks representation illustrate the following observations:

- It is possible to improve the performance of segment-based tracking by automatic resegmentation of the underlying road network representation. Experiments with three resegmentation algorithms demonstrate this and offers insight into which types of modification are most effective in reducing the number of updates.
- It is indeed attractive to use precomputed routes for the moving objects in segment-based tracking, instead of using segments from the road network representation. The GPS logs used confirm conventional wisdom that mobile users are creatures of habit (or efficiency) that frequently use the same routes through the road network to reach their destinations.
- The GPS data used also reveal distinctive speed patterns for some routes and mobile users. The experimental results show that the use of acceleration profiles is capable of increasing the performance of segment-based tracking.

13.8 Further Reading

We proceed to offer an overview of related developments in the commercial and academic arenas. We first offer an overview of 26 tracking-related products and services that we believe are representative of the current commercial state of the art. We then provide an overview of related works within the academic community that may impact future commercial offerings.

13.8.1 Commercially Available Products and Services

Table 13.1 summarizes pertinent properties of what we believe is a representative range of commercially available tracking solutions. The table captures properties of 26 solutions provided by 23 companies. The information that went into the creation of the table was obtained via the Internet during January 2006.

The first column lists company names, and the second lists product names. Starting from the third, each column concerns one product property, and a check mark in a cell indicates that the product in the row of the cell possesses the property corresponding to the column of the cell. The absence of a check mark indicates the opposite.

Columns *GPS*, *Cell*, and *WAAS* concern the means of positioning supported, with *Cell* denoting cellular network-based positioning and *WAAS* denoting the wide area augmentation system that is based on GPS, but offers higher accuracy than GPS by

Table 13.1. Properties of tracking solutions

company	product	GPS	cell	WAAS	SMS	GPRS/CDPD	satellite	phone	custom	time based	on request	sensors	spatial events
BSM	Sentinel	√		√		√				√	√	√	√
Cellfind	look4me		√		√			√			√		
Cybit	mapAmobile		√		√			√			√		
Datafactory	Compact		√					√	√	√			
	Fleetec	√				√				√	√	√	√
Euman	LifePilot	√				√			√	√			
Fleetella	FL1700	√				√			√	√	√		
FleetOnline	FleetOnline		√		√			√			√		
	Trimtrac	√		√					√	√	√		
Global Tracking Solutions	GTS-1000	√				√				√	√	√	√
	Sat-TDiS	√					√			√	√	√	√
GPS Fleet Solutions	Marcus	√				√				√	√	√	√
Gpsnext	Stealth tracker	√				√				√	√		
Guard Magic	VS, VG	√				√	√	√		√			
Mapbyte	Mapaphone		√		√			√			√		
Mobile knowledge	9000 MDT	√				√			√	√	√	√	√
Metro online	AVL	√				√				√	√		
Mobitrac	Mobitrac	√				√	√			√	√		√
Siemens	m.traction Senior Care Service		√		√					√	√		
Telus	Action Tracker	√				√		√		√			
uLocate	fleeTracker	√				√		√		√			
Unteh	Mobitrack	√				√		√		√			
Veriloaction	VL-Tracer	√				√				√	√		
Vettro	Vettro GPS	√				√		√		√			
Web Tech wireless	WebTech5000	√				√				√	√	√	√
2020 Fleet Management	Sentinel Live	√				√			√	√	√		

using corrections. The next three columns concern the types of communication supported, with *SMS* denoting the short messaging service, *GPRS/CDPD* denoting general packet radio service (GSM based) and cellular digital packet data, and *Satellite* denoting satellite-based communications. Then two columns follow that capture the types of terminals supported, with *Phone* denoting mobile phones and *Custom* denoting custom terminals. Finally, the last four columns characterize the type of tracking or how position updates are generated. Here, *Time-based* denotes time-based tracking, that is, updates are issued at regular time intervals; *On request* means that positions of moving objects are pulled from the clients only on request; *Sensors* means that position updates can be generated according to input from external sensors, for example, an alarm, a speedometer reading, a thermometer reading; and *Spatial events*

means that position update can be generated by the moving object entering or leaving a certain region, for example, when leaving the city limits or a prespecified route or when getting into a certain range of a point of interest.

It should be noted that none of the products described in Table 13.1 provides efficient accuracy guarantees or support accuracy-based tracking. Advanced options such as *Spatial events* are usually supported by solutions involving large, custom-made terminals.

13.8.2 Related Academic Contributions

When predicting the future position of an object, the notion of a *trajectory* is typically used [12, 17, 25], where a trajectory is defined in a three-dimensional [17] or four-dimensional [21] space. The dimensions are a two-dimensional "geographical" space, a time dimension, and (possibly) an uncertainty threshold dimension. A point in this space then indicates, for a point in time, the location of an object and the uncertainty of the location. Such points may be computed using speed limits and average speeds on specific road segments belonging to a trajectory.

Wolfson et al. [25] have recently investigated how to incorporate travel-speed prediction in a database. They assume that sensors that can send up-to-date speed information are installed in the roads, and they use average real-time speeds reported every 5 minutes by such in-road sensors. This contrasts the techniques covered in this chapter that use GPS records (termed floating-car data) received from an individual object for predicting that object's movement.

Wolfson et al. [23] propose tracking techniques that offer accuracy guarantees. These assume that objects move on predefined routes already known to the objects, and route selection is done on the client side. If an object changes its route, it sends a position update with information about the new route to the server. The techniques described in this chapter go further by accommodating objects with memory restrictions, and they also work in cases where routes are not known or where map matching fails.

Lam et al. [13] present an adaptive monitoring method that takes into consideration the update, deviation, and uncertainty costs associated with tracking. The method also takes into account the cost of providing incorrect results to queries, during the process of determining when to issue updates. With this method, the moving objects that fall into a query region need close monitoring, and a small accuracy threshold is used for them. Objects not inside a query region may have big thresholds. The techniques presented in this chapter are applicable to this scenario, as they allow different objects to have different thresholds and allow thresholds to change dynamically.

A proposal for trajectory prediction by Karimi and Liu [12] assigns probabilities to the roads emanating from an intersection according to how likely it is that an object entering the intersection will proceed on them. The subroad network within a circular area around an object is extracted, and the most probable route within this network is used for prediction. When the object leaves the current subnetwork, a new subnetwork is extracted, and the procedure is repeated. In this proposal, the

probabilities are global, in that the same probabilities are used for all objects; and they are history-less, in that past choices by an object during a trip are not taken into account when computing probabilities for an object.

Next, Wolfson and Yin [24] consider tracking with accuracy guarantees. Based on experiments with synthetic data, generated to resemble real movement data, they conclude that a version of the point-based tracking is outperformed by a technique that resembles basic segment-based tracking (covered in Sect. 13.3). For a small threshold of 80 m, the latter is a bit more than twice as good as the former; for larger thresholds, the difference decreases. Their dependent variable is numbers of updates per distance unit.

It should also be noted that Ding and Güting [6] have recently discussed the use of what is essentially segment-based tracking within an envisioned system based on their own proposal for a data model for the management of road network constrained moving objects.

When only low accuracy of predicted positions are needed, cellular techniques [1, 14, 15] may be used. With such techniques, the mobile network tracks the cells of the mobile objects in real time to be able to deliver messages or calls to the objects. In this approach, update is handled in the mobile network. In contrast to these techniques, this chapter assumes scenarios where higher accuracies, well beyond those given by the cells associated with the base stations in a cellular network, are needed and where positioning with respect to a road network is attractive.

References

1. Akyildiz IF, Ho JSM (1995) A mobile user location update and paging mechanism under delay constraints. ACM-Baltzer Journal of Wireless Networks 1:244–255
2. Brilingaitė A, Jensen CS, Zokaitė N (2004) Enabling routes as context in mobile services. In: Proceedings of the 12th ACM International Workshop on Geographic Information Systems, 127–136
3. Chung JD, Paek OH, Lee JW, Ryu KH (2002) Temporal pattern mining of moving objects for location-based services. In: Proceedings of the 13th International Conference on Database and Expert Systems Applications, 331–340
4. Čivilis A, Jensen CS, Nenortaite N, and Pakalnis S (2004) Efficient tracking of moving objects with precision guarantees. In: Proceedings of the International Conference on Mobile and Ubiquitous Systems: Networking and Services, 164–173. Extended version available as DB-TR-5, www.cs.aau.dk/DBTR/DBPublications/DBTR-5.pdf
5. Čivilis A, Jensen CS, Pakalnis S (2005) Techniques for efficient road-network-based tracking of moving objects. IEEE Transactions on Knowledge and Data Engineering, 17(5): 698–712
6. Ding Z, Güting RH (2004) Managing moving objects on dynamic transportation networks. In: Proceedings of the 16th International Conference on Scientific and Statistical Database Management, 287–296
7. GloVentures (2006) Glofun Games. www.glofun.com
8. Groundspeak (2006) Geocaching. www.geocaching.com
9. Güting RH, Schneider M (2005) Moving objects databases. Morgan Kaufman

10. Jensen CS, Lahrmann H, Pakalnis S, Runge J (2004) The INFATI Data. Aalborg University, TimeCenter TR-79. www.cs.aau.dk/TimeCenter
11. Jensen CS, Lee K-J, Pakalnis S, Šaltenis S (2005) Advanced tracking of vehicles. In: Proceedings of the Fifth European Congress and Exhibition on Intelligent Transport Systems June 1–3, Hannover, Germany, 12 pages
12. Karimi HA, Liu X (2003) A predictive location model for location-based services. In: Proceedings of the Eleventh ACM International Symposium on Advances in Geographic Information Systems, 126–133
13. Lam K-Y, Ulusoy O, Lee TSH, Chan E, Li G (2001) An efficient method for generating location updates for processing of location-dependent continuous queries. In: Proceedings of the Seventh International Conference on Database Systems for Advanced Applications, 218–225
14. Li G, Lam K-Y, Kuo T-W (2001) Location update generation in cellular mobile computing systems. In: Proceedings of the 15th International Parallel & Distributed Processing Symposium, 96
15. Naor Z, Levy H (1998) Minimizing the wireless cost of tracking mobile users: an adaptive threshold scheme. In: Proceedings of the Seventh Annual Joint Conference on Computer Communications, 720–727
16. Nielsen OA, Jovicic G (2003) The AKTA road pricing experiment in Copenhagen. In: Proceedings of the Tenth International Conference on Travel Behaviour Research Procceedings session 3.2, Lucerne, Switzerland
17. Trajcevski G, Wolfson O, Zhang F, Chamberlain S (2001) The geometry of uncertainty in moving objects databases. In: Proceedings of the Eighth International Conference on Extending Database Technology, 233–250
18. Voisard A, Schiller J (2004) Location-based Services. Morgan Kaufmann
19. Wikipedia (2006) Galileo positioning system. en.wikipedia.org/wiki/Galileo_positioning_system
20. Wikipedia (2006) Global positioning system. en.wikipedia.org/wiki/GPS
21. Wolfson O (2001) The opportunities and challenges of location information management. In: tech. report presented at Comp. Science and Telecommunications Board Workshop on Intersections of Geospatial Information and Information Technology. www7.nationalacademies.org/cstb/wp_geo_wolfson.pdf
22. Wolfson O, Chamberlain S, Dao S, Jiang L, Mendez G (1998) Cost and imprecision in modeling the position of moving objects. In: Proceedings of the Fourteenth International Conference on Data Engineering, 588–596
23. Wolfson O, Sistla AP, Camberlain S, Yesha Y (1999) Updating and querying databases that track mobile units. Distributed and Parallel Databases 7(3):257–287
24. Wolfson O, Yin H (2003) Accuracy and resource consumption in tracking and location prediction. In: Proceedings of the Eight International Symposium on Spatial and Temporal Databases 325–343
25. Xu B, Wolfson O (2003) Time-series prediction with applications to traffic and moving objects databases. In: Proceedings of the Third ACM International Workshop on Data Engineering for Wireless and Mobile Access, 56–60

References containing URLs are valid as of May 29th, 2006.