

METHODS AND SYSTEMS FOR PREDICTING JOURNEY TIME

TECHNICAL FIELD

The present disclosure relates to electronic commerce. In particular, the present disclosure relates to the prediction of journey times, for example in delivery or ride booking applications.

BACKGROUND

Many electronic commerce applications require prediction of journey times. For example, ride sharing or taxi booking applications may require estimated time of arrival of a taxi at a pickup point and an estimated time of arrival at a destination. Food delivery applications may require an estimated time of arrival at a drop off point. Journey times are generally dependent on static factors such as the distance between locations, and dynamic factors such as traffic levels on roads. In order to model traffic levels, many applications use live traffic data as an input to systems. This has a number of disadvantages, firstly, a large amount of real-time processing is required to predict journey times using live traffic data. Secondly, such systems require the live traffic data as an input, in some locations live traffic data may not be available.

SUMMARY

According to a first aspect of the present disclosure, a method of predicting journey time is provided. The method comprises: obtaining a set of initial speed profiles, each initial speed profile of the set of initial speed profiles corresponding to a road portion on an initial map of a geographic area and comprising a set of speed values for the road portion, each speed value corresponding to a time interval; receiving a plurality of global positioning system data sets, each global positioning system data set corresponding to a respective journey made by a respective vehicle in the geographic area and comprising a set of geolocation indications and corresponding time indications; determining speed data from the of geolocation indications and corresponding time indications corresponding to journey segments along respective road portions; updating initial speed profiles of the set of initial speed profiles using the

speed data to determine a set of current speed profiles; training a journey model using at least the set of current speed profiles; and determining a predicted journey time for a journey using the journey model.

The present disclosure provides methods of predicting journey time and giving an accurate estimated time of arrival without the need for live traffic information. Instead of requiring live traffic information, speed profiles for roads or road sections are used.

In an embodiment obtaining the set of initial speed profiles comprises looking up a set of previously updated speed profiles as the initial set of speed profiles. The speed profiles may be based on historical data. As new data is generated, the speed profiles may be updated. It is noted that the speed profiles may be updated on a regular basis as new data becomes available. During the update process, it is only the new data that requires processing since the previous version of the speed profiles depends on the earlier data.

In an embodiment, obtaining the set of initial speed profiles comprises generating a set of global speed profiles for the road portions according to a road type associated with each of the road portions. Global speed profiles may be generated for different road types. These global speed profiles may be updated with data for specific roads to allow specific speed profiles to be generated.

In an embodiment the method further comprises mapping the set of current speed profiles to a current map of the geographic area. If there is a change in road layout or other change in mapped features, the speed profiles may be adapted by mapping the speed profiles to the new, current map of the geographic area. For example, the speed profiles may be mapped to an index rather directly to mapping data. This allows previous versions of the speed profiles that relate to previous map versions to be re-mapped to an updated map without the requirement to completely regenerate all of the speed profiles.

In an embodiment, determining speed data from the geolocation indications and corresponding time indications corresponding to journey segments along the respective road portions comprises determining a double median absolute variation.

In an embodiment, the method further comprises filtering the global positioning system data sets. This may comprise computing an accuracy threshold and discarding any geolocation indications and corresponding time indications that are below the accuracy threshold. The accuracy threshold may be calculated from a mean or median accuracy of a global positioning system data set.

According to a second aspect of the present disclosure, a computer readable carrier medium carrying processor executable instructions operable to cause a processor to carry out a method set out above is provided.

According to a third aspect of the present disclosure, a data processing system for predicting journey time is provided. The data processing system comprises a processor, and program storage, the program storage storing computer program instructions operative by the processor to: obtain a set of initial speed profiles, each initial speed profile of the set of initial speed profiles corresponding to a road portion on an initial map of a geographic area and comprising a set of speed values for the road portion, each speed value corresponding to a time interval; receive a plurality of global positioning system data sets, each global positioning system data set corresponding to a respective journey made by a respective vehicle in the geographic area and comprising a set of geolocation indications and corresponding time indications; determine speed data from the of geolocation indications and corresponding time indications corresponding to journey segments along respective road portions; update initial speed profiles of the set of initial speed profiles using the speed data to determine a set of current speed profiles; train a journey model using at least the set of current speed profiles; and determine a predicted journey time for a journey using the journey model.

In an embodiment, the program storage stores computer program instructions operative by the processor to: obtain the set of initial speed profiles by looking up a set of previously updated speed profiles as the initial set of speed profiles.

In an embodiment, the program storage stores computer program instructions operative by the processor to: obtain the set of initial speed profiles by generating a

set of global speed profiles for the road portions according to a road type associated with each of the road portions.

In an embodiment, the program storage stores computer program instructions operative by the processor to: map the set of current speed profiles to a current map of the geographic area.

In an embodiment, the program storage stores computer program instructions operative by the processor to: determine speed data from the geolocation indications and corresponding time indications corresponding to journey segments along the respective road portions by determining a double median absolute variation.

In an embodiment, the program storage stores computer program instructions operative by the processor to: filter the global positioning system data sets.

In an embodiment, the program storage stores computer program instructions operative by the processor to: filter the global positioning system data sets by computing an accuracy threshold and discarding any geolocation indications and corresponding time indications that are below the accuracy threshold.

In an embodiment, the accuracy threshold calculated from a mean or median accuracy of a global positioning system data set.

BRIEF DESCRIPTION OF THE DRAWINGS

In the following, embodiments of the present invention will be described as non-limiting examples with reference to the accompanying drawings in which:

FIG.1A and FIG.1B are bar charts showing speed profiles used in embodiments of the present invention;

FIG.2 is a block diagram showing a journey time prediction system according to an embodiment of the present invention;

FIG.3 is a flow chart showing a method of generating global speed profiles used in embodiments of the present invention; and

FIG.4 is a flow chart showing a method of predicting a journey time according to an embodiment of the present invention.

DETAILED DESCRIPTION

The present disclosure relates to predicting journey times for example in food delivery or ride booking applications. The methods of the present disclosure use speed profiles which are roads, road sections, or types of road to predict journey times. The speed profiles may be generic or global speed profiles which are associated with a road type, for example, “highway” or “residential road”, the speed profiles may also be associated with specific roads or road sections.

FIG.1A and FIG.1B are bar charts showing speed profiles used in embodiments of the present invention.

As shown FIG.1A and FIG.1B, the speed profiles indicate speeds associated with one hour periods, and the speed profile shown in FIG.1A indicates a speed of 22km/h for the time period 00:00 to 01:00. Similarly, the speed profile shown in FIG.1B indicates a speed of 62km/h for the time period 00:00 to 01:00. As will be explained in more detail below, the speeds indicated by the speed profiles are the average speeds of vehicles on the road associated with the speed profile during the corresponding time periods.

As can be seen from FIG.1A and FIG.1B, the speeds of the road or road type corresponding to FIG.1A are generally lower than the speeds of the road or the road type corresponding to FIG.1B. However, there is a greater drop in speed between 08:00 and 10:00, and also between 16:00 and 19:00 shown in FIG.1B. This may be due to FIG.1B being associated with a highway on which the speed decreases during morning and evening rush hour. FIG.1A may be associated with a residential road which generally has slower speeds which are not strongly affected by morning and evening rush hours.

FIG.2 is a block diagram showing a journey time prediction system according to an embodiment of the present invention. The journey time prediction system 100 shown in FIG.2 uses speed profiles such as those shown in FIG.1A and FIG.1B to predict journey times. As described in more detail below, the processing carried out by the journey time prediction system 100 may involve generating and / or updating speed profiles.

The journey time prediction system 100 comprises a processor 110, a working memory 112, a network interface 114, program storage 120, and data storage 130. The processor 110 may be implemented as one or more central processing unit (CPU) chips. The program storage 120 is a non-volatile storage device such as a hard disk drive which stores computer program modules. The computer program modules are loaded into the working memory 112 for execution by the processor 110. The data storage 130 is a non-volatile storage device which stores data which is used by the journey time prediction system 100 during processing. The data storage 130 may be implemented as a number of separate data storage devices. These devices may physically form part of the journey time prediction system 100 or may be separate systems such as servers which communicate with the journey time prediction system 100 to provide data when required. The network interface 114 is an interface that allows the journey time prediction system 100 to communicate with other devices. These other devices may comprise for example devices associated with merchants to allow merchant data to be entered or selections to be made by the merchants.

The program storage 120 stores a global speed profile calculation module 121, a global positioning system (GPS) data processing module 122, a speed calculation module 123, a speed profile update module 124, a journey model training module 125, and a routing engine module 126. The computer program modules cause the processor 110 to execute various processing methods which are described in more detail below. The program storage 120 may be referred to in some contexts as computer readable storage media and/or non-transitory computer readable media. As depicted in FIG.2, the computer program modules are distinct modules which perform respective functions implemented by the journey time prediction system 100. It will be appreciated that the boundaries between these modules are exemplary only, and that alternative embodiments may merge modules or impose an alternative decomposition

of functionality of modules. For example, the modules discussed herein may be decomposed into sub-modules to be executed as multiple computer processes, and, optionally, on multiple computers. Moreover, alternative embodiments may combine multiple instances of a particular module or sub-module. It will also be appreciated that, while a software implementation of the computer program modules is described herein, these may alternatively be implemented as one or more hardware modules (such as field-programmable gate array(s) or application-specific integrated circuit(s)) comprising circuitry which implements equivalent functionality to that implemented in software.

Although the journey time prediction system 100 is described with reference to a computer, it should be appreciated that the journey time prediction system 100 may be formed by two or more computers in communication with each other that collaborate to perform a task. For example, but not by way of limitation, an application may be partitioned in such a way as to permit concurrent and/or parallel processing of the instructions of the application. Alternatively, the data processed by the application may be partitioned in such a way as to permit concurrent and/or parallel processing of different portions of a data set by the two or more computers. In an embodiment, virtualization software may be employed by the journey time prediction system 100 to provide the functionality of a number of servers that is not directly bound to the number of computers in the journey time prediction system 100. In an embodiment, the functionality disclosed above may be provided by executing the application and/or applications in a cloud computing environment. Cloud computing may comprise providing computing services via a network connection using dynamically scalable computing resources. A cloud computing environment may be established by an enterprise and/or may be hired on an as-needed basis from a third party provider.

The data storage 130 stores speed profile data 132, map data 136, booking data 140, driver data 142, geographic information system (GIS) data 144, journey global positioning system (GPS) data 146 and journey models 150.

The speed profile data 132 comprises global speed profiles 133, specific speed profiles 134 and speed profile mapping data 135. The global speed profiles 133 are speed profiles such as those illustrated in FIG.1A and FIG.1B which are associated

with road types, such as highways or residential roads. There may be, for example, 24 different road types with a global speed profile associated with each different road type. The specific speed profiles 134 are speed profiles which such as those illustrated in FIG.1A and FIG.1B are associated with specific roads or road sections. The speed profile mapping data 135 maps the specific speed profiles 134 to locations on map data which is stored as map data 136. The specific speed profiles 134 may be stored associated with an index rather than a specific location and the speed profile mapping data 135 indicates the map locations which correspond to the road to which the specific speed profile corresponds. The reason for storing the specific speed profiles 134 and speed profile mapping data 135 in this way is that it provides robustness against map changes. For example, if there is a change in road layout, then the roads with which parts of the specific speed profiles 134 are associated may change, and this change can be taken into account by updating the speed profile mapping data 135 without the requirement to update all of the specific speed profiles 134. The global speed profiles 133 and the specific speed profiles 134 may be stored as vectors or matrices with a speed associated with each one hour period. However, speed profiles with different periods may be used, so a day may be divided in to 24 one hour periods or a different number of periods. In addition, separate speed profiles may be stored for weekdays and weekends or holidays to account for differing traffic levels between different types of day.

The map data 137 comprises previous map data 137 and current map data 138. As discussed above, when features such as road layouts change, new map data may be generated. Different versions of map data 137 may be stored to take into account such changes.

The booking data 140 stores data of completed orders such as taxi journeys or deliveries. This may include details of the time of day, day of the week, whether the day was a public holiday, details of the driver that carried out the booking, pickup and drop off locations, and details of the journey. The driver data 142 stores details of drivers registered to use an online booking or delivery system, this may include details of the driver, details of bookings or orders carried out by the driver and details of their vehicle. The geographic information system (GIS) data 144 stores protocol buffer binary format (PBF) information such as information about roads, and other descriptive

information about roads or geographic features. The journey global positioning system (GPS) data 146 stores GPS data from journeys made by drivers. The journey GPS data 146 is used in various processing by the journey time prediction system 100 to generate speed profiles for use in predicting journey times.

The journey models 150 are machine learning models for journeys. There may be multiple journey models depending on the use case. For example, in a taxi booking system, there may be a journey model for journeys such as travelling to a pick up point to collect a passenger and a separate journey model for the journey transporting the passenger to their destination. In a for delivery scenario, again, there may be multiple separate journey models. The journey models may be any form of regression-based machine learning models. The training of the journey models is discussed in more detail below in relation to step 412 shown in FIG.4.

FIG.3 is a flow chart showing a method of generating global speed profiles used in embodiments of the present invention. The method 300 shown in FIG.3 is carried out by the global speed profile calculation module 121 of the journey time prediction system 100 shown in FIG.2.

The method 300 shown in FIG.3 optimizes the global speed profiles 133 of the speed profile data 132 stored in the data storage 130. The optimization of global speed profiles helps with the later processing which optimizes the specific speed profiles 134.

In step 302, the global speed profile calculation module 121 is executed by the processor 110 of the journey time prediction system 100 to generate default speed profiles. Default speed profiles are generated for each road type with a speed for each one hour period.

In step 304, the global speed profile calculation module 121 is executed by the processor 110 of the journey time prediction system 100 to extract journey times from the booking data 140 stored in the data storage 130. This may involve extracting historical booking data for an appropriate vehicle type such as a four wheeled vehicle or a two wheeled vehicle. The trip time as the time between pickup and drop off is then extracted from the booking data 140 stored in the data storage 130. The different

global speed profiles may be calculated for different vehicles and for different types of day, for example separate speed profiles may be calculated for weekdays and weekends. Speed profiles also be calculated for specific cities.

In step 306, the global speed profile calculation module 121 is executed by the processor 110 of the journey time prediction system 100 to optimize the road type speed profiles. An optimization algorithm such as Bayesian optimization is run. The goal is to find the optimal speed profiles within a given bound for each hour covering all road type. The processing is done in parallel for each one hour period for all regions to that are to be optimized. This optimizes road type speed profiles globally.

In step 308, the global speed profile calculation module 121 is executed by the processor 110 of the journey time prediction system 100 to save the global speed profiles 133 in the speed profile data 132 of the data storage 130.

FIG.4 is a flow chart showing a method of predicting a journey time according to an embodiment of the present invention. The method 400 shown in FIG.4 is carried out by the journey time prediction system 100 shown in FIG.2.

In step 402, speed profile update module 124 is executed by the processor 110 of the journey time prediction system 110 to obtain initial speed profiles. In some implementations, the initial speed profiles are the global speed profiles 133 stored in the speed profile data of the data storage 130. In other implementations, the initial speed profiles are specific speed profiles 134 that have previously been updated. In both cases, the method 400 involves updating the initial speed profiles using GPS data for recent journeys. The initial speed profiles each correspond to a road portion of the map data 136, in the case of specific speed profiles, they may correspond to previous map data 137. That is, the previously updated speed profiles may relate to an old map version. In the case of global speed profiles, the GIS data 144 may be used to determine road types and then a corresponding global speed profile for that road type may be taken as the initial speed profile for a given road section.

In step 404, the network interface 114 of the journey time prediction system 110 receives global positioning (GPS) data sets. The GPS data sets comprise sets of GPS

pings corresponding to a respective journey made by a respective vehicle in the geographic area and comprising a set of geolocation indications and corresponding time indications.

In step 406, the GPS data processing module is executed by the processor 110 of the journey time prediction system 100 to filter the GPS data sets. The GPS data sets are snapped to a road network graph. The GPS pings are processed using several heuristics. For example, an intra GPS pair should be close in temporal and spatial distance. Then then score at ping level (considering the both heuristic as well as given GPS accuracy) if it is a “good” or “bad” ping. The “bad” pings are then filtered out. The processing may take place as follows: i) compute the time difference between ping intervals; ii) compute distance difference between ping intervals; iii) score each ping using a weighted average based on the threshold of time difference, distance difference as well as GPS accuracy. If score is below a threshold, it is determined that the ping is a “bad” ping it is discarded from further processing. The filtered raw pings are mapped and snapped onto a graph network using a map-matching algorithm from a routing engine. In order to optimize the process, the ping data are broken into partitions to reduce routing engine time-out; the median / mean GPS accuracy of a given booking and the values are clipped. For example, a level of 25 may be taken as the threshold for clipping and values below the threshold are clipped. Map-matching is performed to map GPS pings to road network, then route assessment is performed by comparing raw GPS ping polylines against matched pings polylines using Hausdorff and Frechet distance. Pings that exceed a threshold distance from the polyline are filtered off.

In step 408, the speed calculation module is executed by the processor 110 of the journey time prediction system 100 to determine speed data from the of geolocation indications and corresponding time indications corresponding to journey segments along respective road portions. Speed data is determined from the filtered GPS pings. Speed is calculated using distance between the first and last ping of a given edge divided by the total time. Thus, speed is computed using the mapped GPS data. Double median absolute deviation may be performed to filter out outlier speeds computed for a given hour for a given edge.

In step 410, the speed profile update module 124 is executed by the processor 110 of the journey time prediction system 100. The speeds calculated in step 408 are used to update the initial speed profiles. Once the speeds are computed and filtered for outliers, the data is aggregated as there may be multiple vehicles on a particular road (edge) at the same time. The mean or, alternatively, the median may be used to aggregate as an option. After aggregation, there is one speed value per road per hourly interval. This is used to update the previously stored speed profile. The residual is determined as the difference between the computed speed for the speed profile and the previous speed profile. A weight is applied to this difference and the weighted difference is then added to the previous value in the speed profile. The weight determines how much influence the newly calculated speed has on the speed profile. For example, if the stored value in the speed profile is 30km/h and the newly calculated speed is 45km/h. In this case, the difference is $45-30=15$. If the weight is 0.1, then the updated speed profile value will be $30 + (15 * 0.1) = 31.5$. It is noted that if there a large change in the speed such as in the example discussed above, over time, the speed profile will gradually converge on the correct value.

Following step 410, a map to map mapping may be carried out. This is because when there is a map layer change, there is a need for the system to understand the change in indexing of edges (i.e. roads) from an arbitrary older version of map to the latest map. Therefore, the journey time prediction system 100 uses two metrics: similarity score and success rate to measure the mapping process. If there is a high similarity score, then it is not necessary to perform a new map to map mapping. If there is a low similarity score, then the map to map mapping is carried out. Pass list of edge indexes along with their coordinates to mapper then iterate through all edge indexes, and given the coordinates, search a specific radius for possible new edge indexes candidates in new map. Greedily match old edge index to the closest new edge index, amongst the returned candidates with the same coordinates and direction. If there is a low success rate, this may indicate an issue with the map and the process is halted. A low success rate is often due to invalid changes made to the map. It is expected that the success rate will be at least 90%. If the success rate is below this, the new map PBF (Protocolbuffer Binary Format) input is checked for inconsistency. Such inconsistencies may be due to incorrect edges having been added, duplicated roads or similar errors.

In step 412, the journey model training module 125 is executed by the processor 110 of the journey time prediction system 100 to train journey models 150 stored in the data storage 130. The booking data 140, driver data 142, GIS data 144 and updated speed profiles are used to train the journey models 150. Based on the processing described above with reference to step 406, the filtered GPS pings snapped onto the road network graph may be stored in a database. The snapped pings can then be used to compute an approximate driver distance. For example, the Haversine distance between the snapped pings may be used. Geohash based inputs may also be calculated which discretize the map into grids of different granularity. The geohash inputs may capture various attributes of a geohash, for example attributes such as the popularity, or busyness of a geohash may be captured from the total number of drivers, customers or primary points of interest (PPOIs) in a given geohash. Since a journey comprises of an origin and a destination, every journey may be described by a pair of geohashes (an origin geohash and a destination geohash). These pairs of geohashes may be used as inputs to the journey training module 125 to train the journey models. Additional inputs are time-based inputs such as time of the day, day of the week, number of days to a holiday/weekend, weekend indicator, holiday indicator.

Tags extracted from the GIS data such as the number of one-way/two-way roads, lane count, surface type, speed limit, highway type may be used to compute aggregated statistics (such as total sum, or count) which can be used to describe a journey. The journey models may be any form of regression-based machine learning models. Typically, 2 months' worth of data is used in the training. The training process may be carried out on a weekly basis using the latest 2 months' worth of data.

Different models may be trained for different countries and the identity of cities may be included as an input which allows the model to distinguish between cities for larger countries e.g. Indonesia. Highway type may be encoded using average historical actual time of arrival conditioned on highway type to train the model. For example, using a certain highway may result in a higher average arrival time than other highways.

In step 414, the routing engine module 126 is executed by the processor 110 of the journey time prediction system 100 to determine a predicted journey time for journey. The journey models 150 trained in step 412 are used with inputs comprising an indication of the driver, the start and end locations of the journey and the time at which the journey is predicted to take place. Based on this input information, the journey models output an indication of the predicted journey time. The journey models 150 may also take in country (this allows the service to determine the appropriate model to use), city (if applicable), the vehicle type as well as journey type as inputs. The service will get the input as described above and will first interact with the routing engine in real time to get additional inputs: Since there is a need to get a possible trajectory from origin to destination, the routing engine will recommend such a path (which is influenced by the speed profiles itself). The “expected” trajectory to the destination will allow computation for inputs like aggregated speed profiles along the recommended route, sequence of GIS tags – encoded and aggregated similar to how it was trained. Once it collated all the inputs as per training, it will then provide the vector to the model.

The predicted journey time may be provided to a user, for example as an indication of an estimated time of arrival of an order, an estimated time of arrival of a booked taxi or an estimated time of arrival of a passenger.

Whilst the foregoing description has described exemplary embodiments, it will be understood by those skilled in the art that many variations of the embodiments can be made within the scope and spirit of the present invention.

CLAIMS

1. A method of predicting journey time, the method comprising:
 - obtaining a set of initial speed profiles, each initial speed profile of the set of initial speed profiles corresponding to a road portion on an initial map of a geographic area and comprising a set of speed values for the road portion, each speed value corresponding to a time interval;
 - receiving a plurality of global positioning system data sets, each global positioning system data set corresponding to a respective journey made by a respective vehicle in the geographic area and comprising a set of geolocation indications and corresponding time indications;
 - determining speed data from the of geolocation indications and corresponding time indications corresponding to journey segments along respective road portions;
 - updating initial speed profiles of the set of initial speed profiles using the speed data to determine a set of current speed profiles;
 - training a journey model using at least the set of current speed profiles; and
 - determining a predicted journey time for a journey using the journey model.
2. A method according to claim 1, wherein obtaining the set of initial speed profiles comprises looking up a set of previously updated speed profiles as the initial set of speed profiles.
3. A method according to claim 1, wherein obtaining the set of initial speed profiles comprises generating a set of global speed profiles for the road portions according to a road type associated with each of the road portions.
4. A method according to any preceding claim, further comprising mapping the set of current speed profiles to a current map of the geographic area.
5. A method according to any preceding claim, wherein determining speed data from the geolocation indications and corresponding time indications corresponding to journey segments along the respective road portions comprises determining a double median absolute variation.

6. A method according to any preceding claim, further comprising filtering the global positioning system data sets.

7. A method according to claim 6, wherein filtering the global positioning system data sets comprises computing an accuracy threshold and discarding any geolocation indications and corresponding time indications that are below the accuracy threshold.

8. A method according to claim 7, wherein the accuracy threshold calculated from a mean or median accuracy of a global positioning system data set.

9. A computer readable carrier medium carrying processor executable instructions operable to cause a processor to carry out a method according to any one of claims 1 to 8.

10. A data processing system for predicting journey time, the data processing system comprising a processor, and program storage, the program storage storing computer program instructions operative by the processor to:

obtain a set of initial speed profiles, each initial speed profile of the set of initial speed profiles corresponding to a road portion on an initial map of a geographic area and comprising a set of speed values for the road portion, each speed value corresponding to a time interval;

receive a plurality of global positioning system data sets, each global positioning system data set corresponding to a respective journey made by a respective vehicle in the geographic area and comprising a set of geolocation indications and corresponding time indications;

determine speed data from the of geolocation indications and corresponding time indications corresponding to journey segments along respective road portions;

update initial speed profiles of the set of initial speed profiles using the speed data to determine a set of current speed profiles;

train a journey model using at least the set of current speed profiles; and

determine a predicted journey time for a journey using the journey model.

11. A data processing system according to claim 10, wherein the program storage stores computer program instructions operative by the processor to: obtain the set of

initial speed profiles by looking up a set of previously updated speed profiles as the initial set of speed profiles.

12. A data processing system according to claim 10, wherein the program storage stores computer program instructions operative by the processor to: obtain the set of initial speed profiles by generating a set of global speed profiles for the road portions according to a road type associated with each of the road portions.

13. A data processing system according to any one of claims 10 to 12, wherein the program storage stores computer program instructions operative by the processor to: map the set of current speed profiles to a current map of the geographic area.

14. A data processing system according to any one of claims 10 to 13, wherein the program storage stores computer program instructions operative by the processor to: determine speed data from the geolocation indications and corresponding time indications corresponding to journey segments along the respective road portions by determining a double median absolute variation.

15. A data processing system according to any one of claims 10 to 14, wherein the program storage stores computer program instructions operative by the processor to: filter the global positioning system data sets.

16. A data processing system according to claim 15, wherein the program storage stores computer program instructions operative by the processor to: filter the global positioning system data sets by computing an accuracy threshold and discarding any geolocation indications and corresponding time indications that are below the accuracy threshold.

17. A data processing system according to claim 16, wherein the accuracy threshold calculated from a mean or median accuracy of a global positioning system data set.