



US 20110208426A1

(19) **United States**

(12) **Patent Application Publication**
Zheng et al.

(10) **Pub. No.: US 2011/0208426 A1**

(43) **Pub. Date: Aug. 25, 2011**

(54) **MAP-MATCHING FOR
LOW-SAMPLING-RATE GPS TRAJECTORIES**

Publication Classification

(75) Inventors: **Yu Zheng**, Beijing (CN); **Yin Lou**,
Shanghai (CN); **Chengyang Zhang**,
Beijing (CN); **Xing Xie**, Beijing
(CN)

(51) **Int. Cl.**
G01C 21/26 (2006.01)

(52) **U.S. Cl.** **701/208**

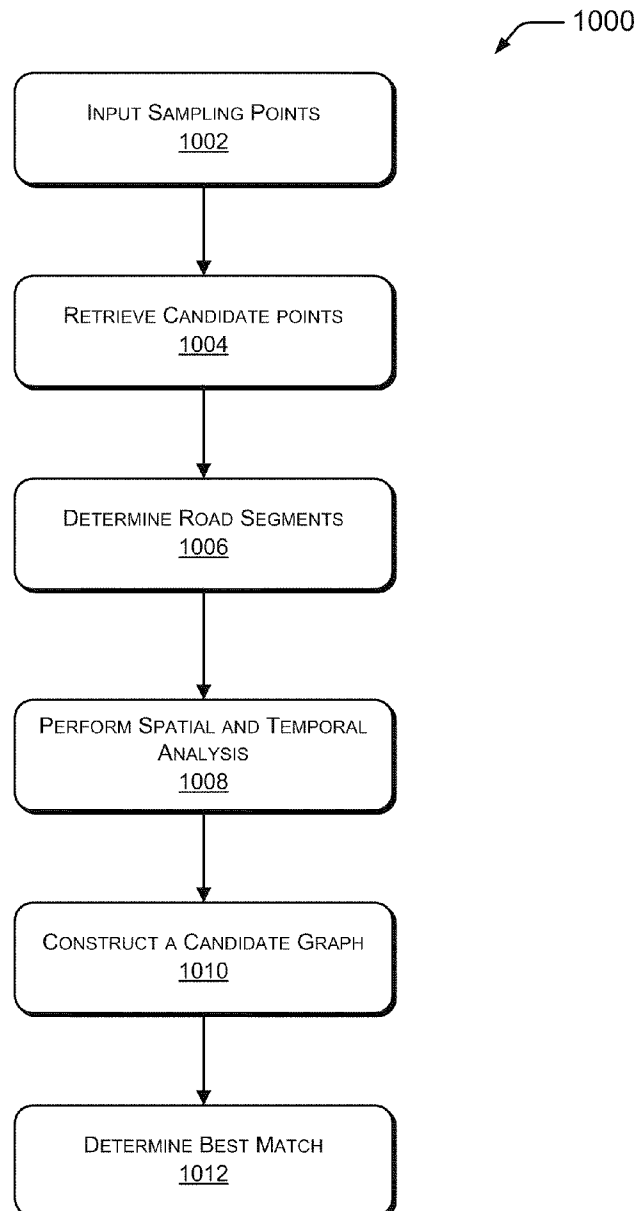
(73) Assignee: **Microsoft Corporation**, Redmond,
WA (US)

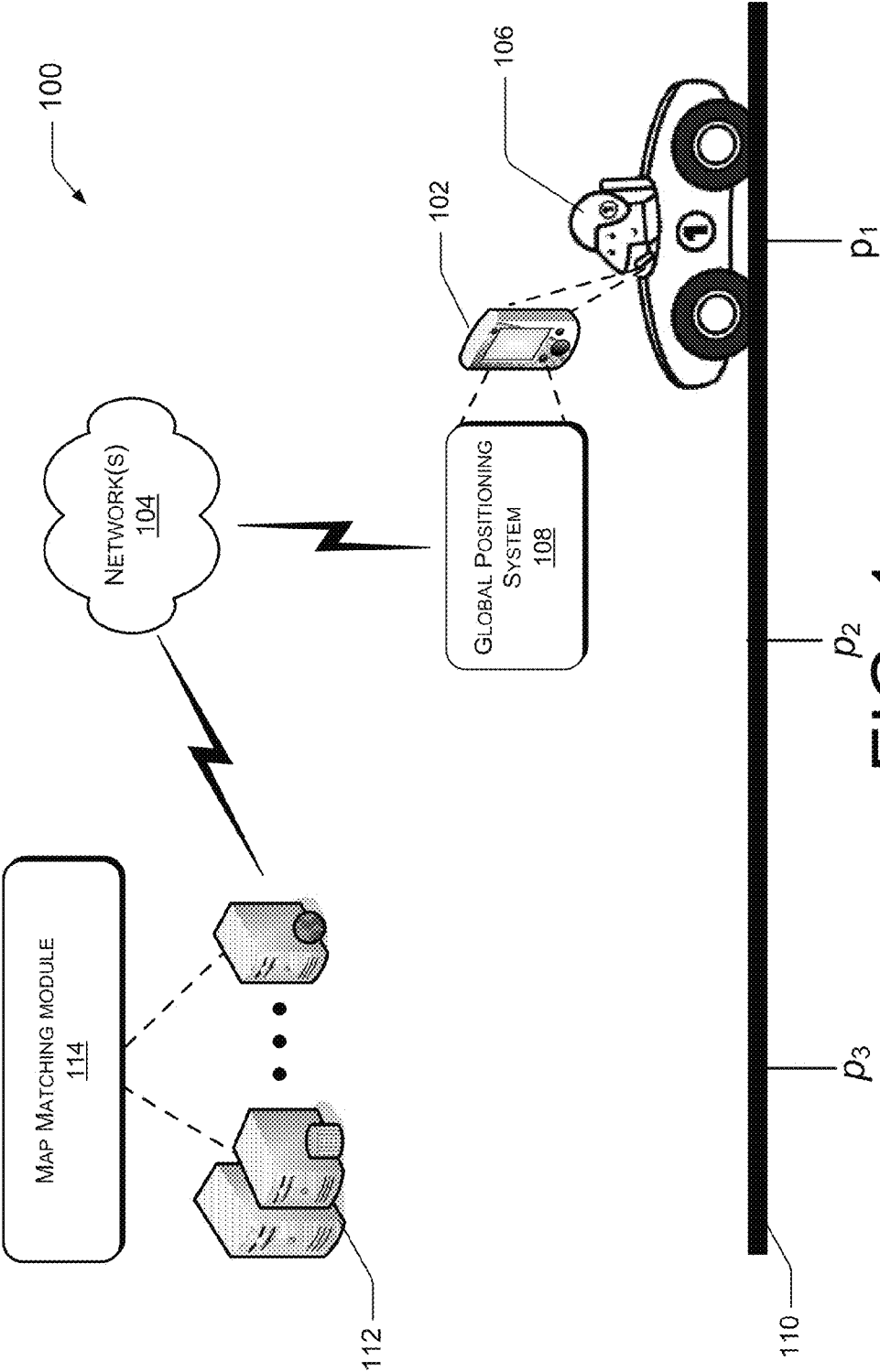
(57) **ABSTRACT**

(21) Appl. No.: **12/712,857**

This disclosure describes a map-matching module that supports a Global Positioning System (GPS) and provides a user with a best match trajectory corresponding to GPS sampling points taken at a low sampling rate. The best match trajectory is based upon a spatial-temporal analysis.

(22) Filed: **Feb. 25, 2010**





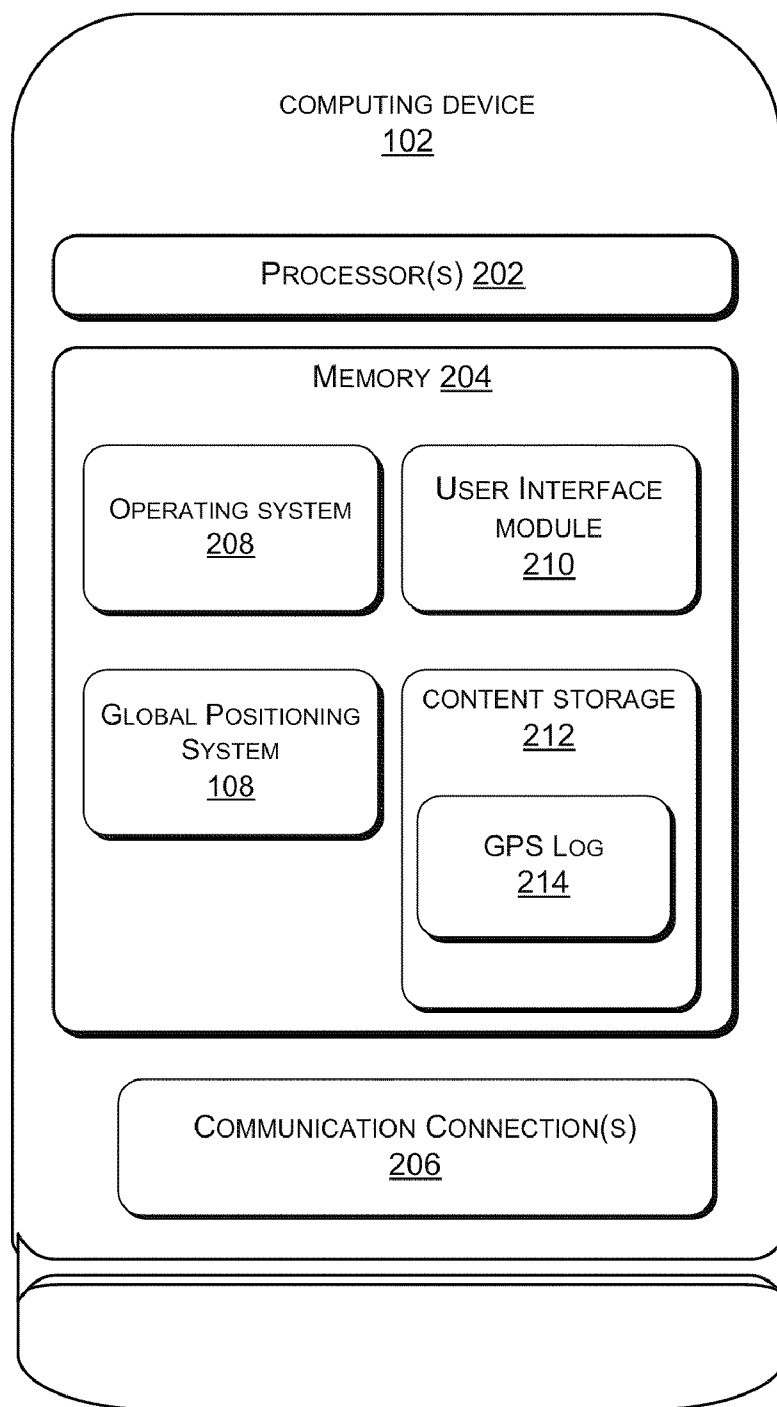


FIG. 2

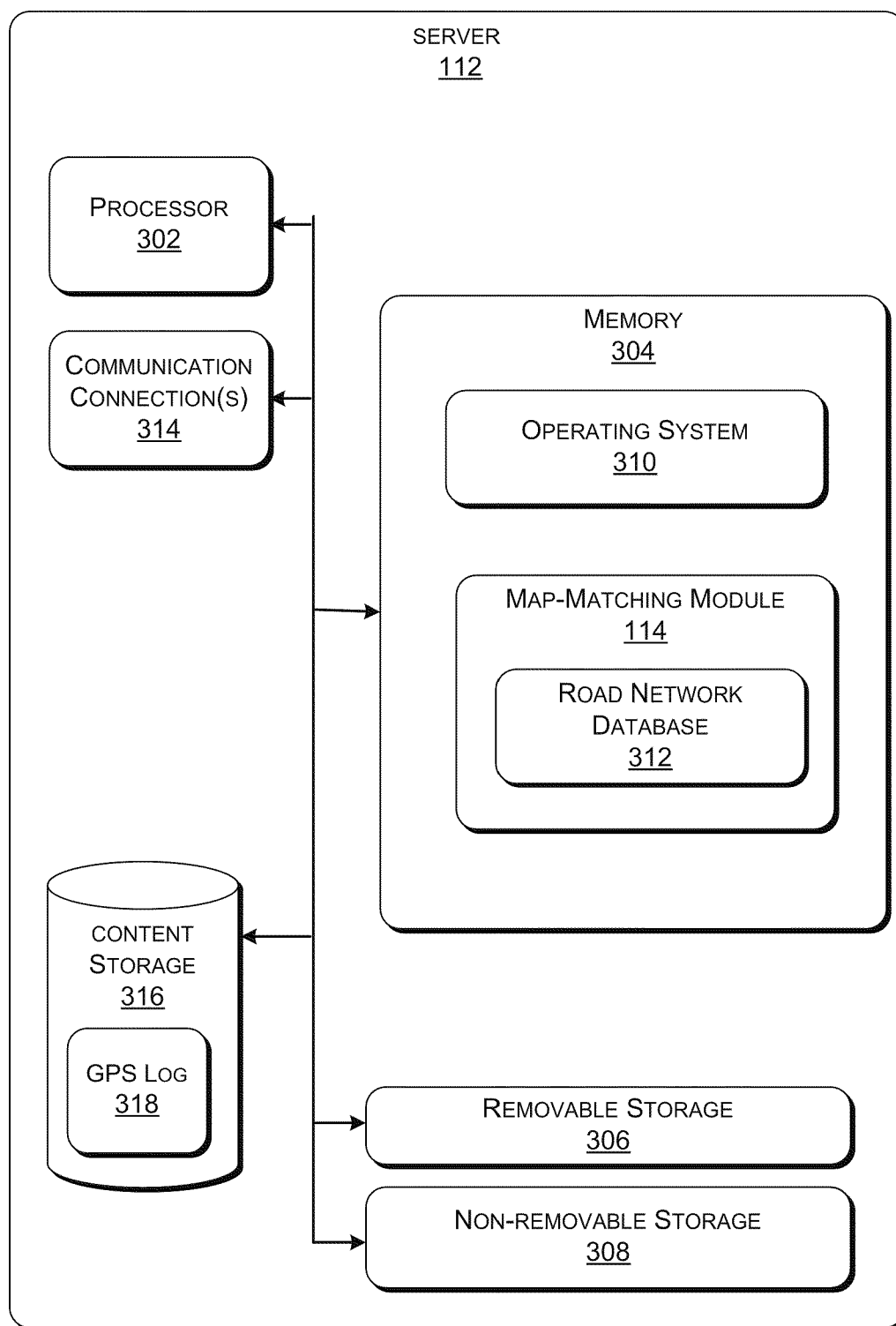


FIG. 3

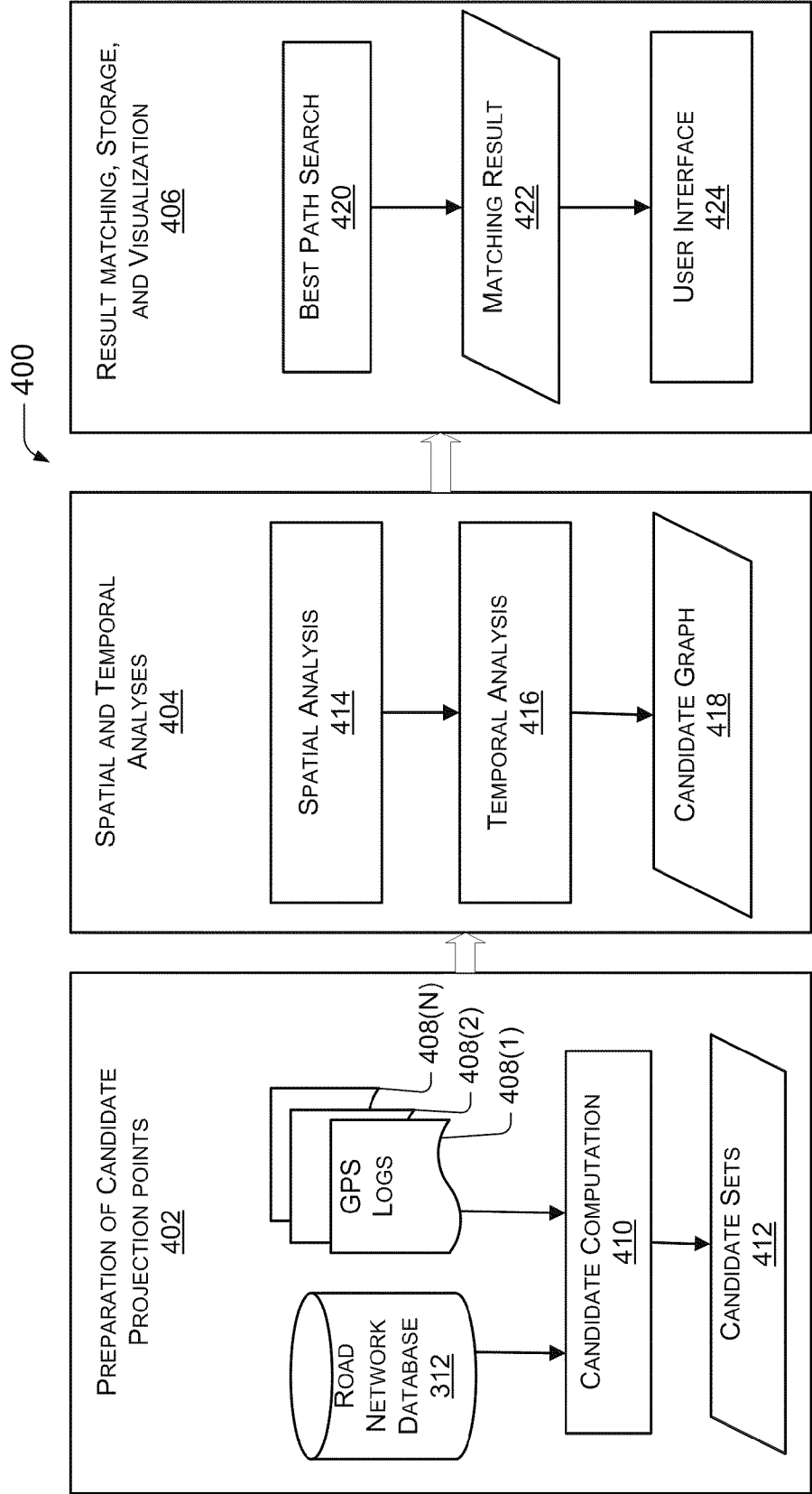


FIG. 4

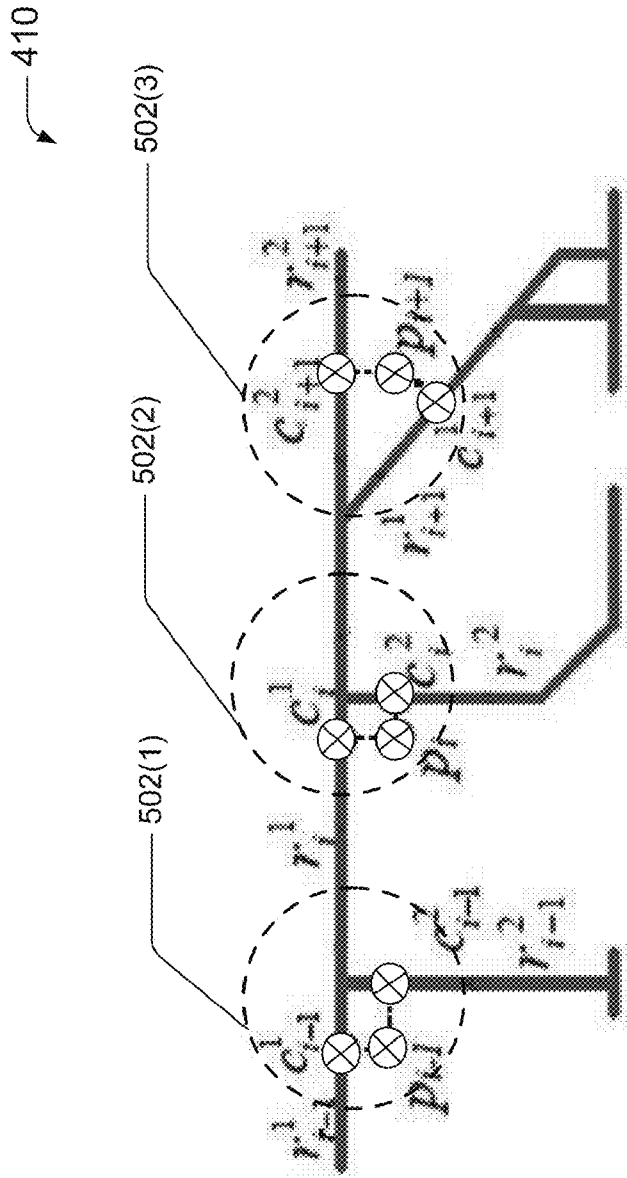


FIG. 5

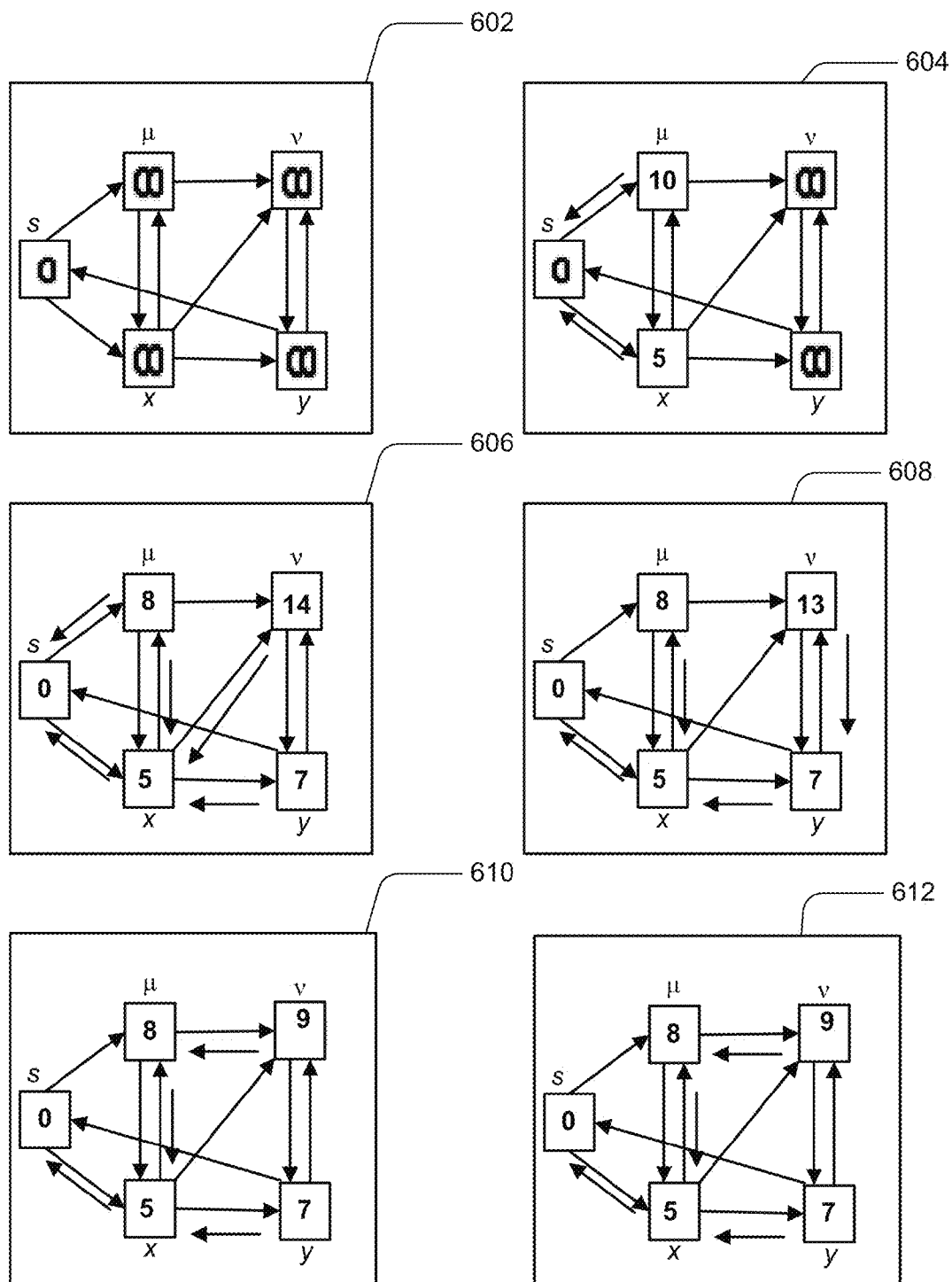


FIG. 6

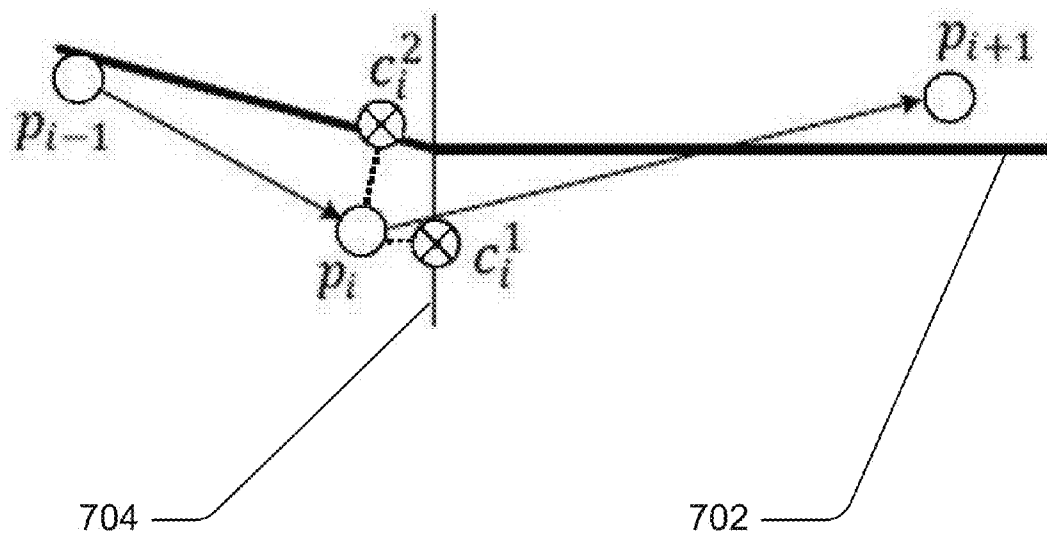


FIG. 7

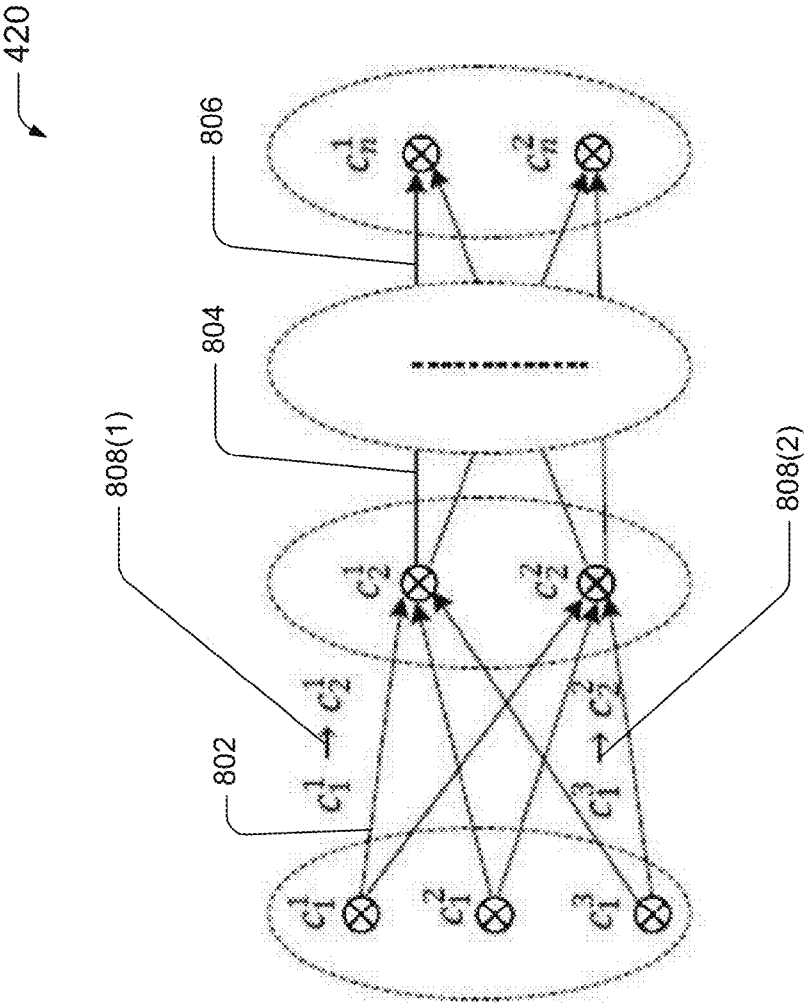


FIG. 8

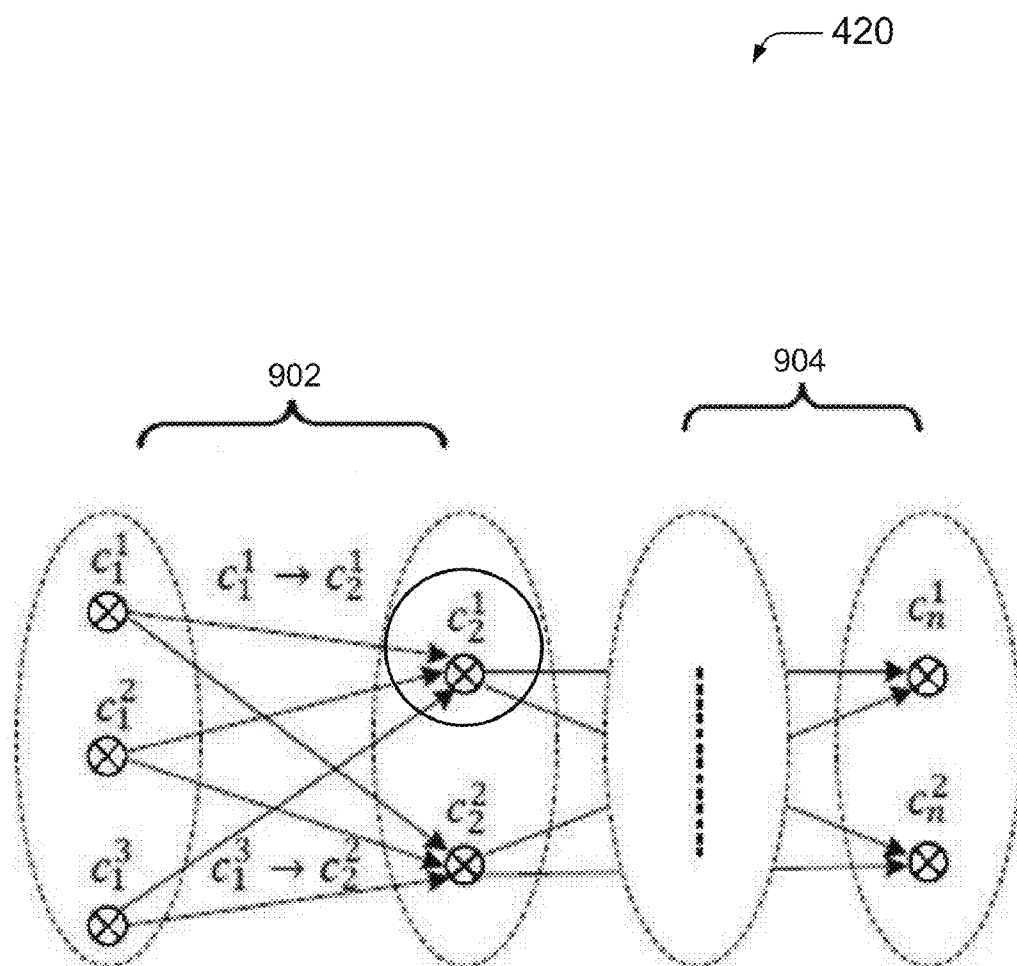


FIG. 9

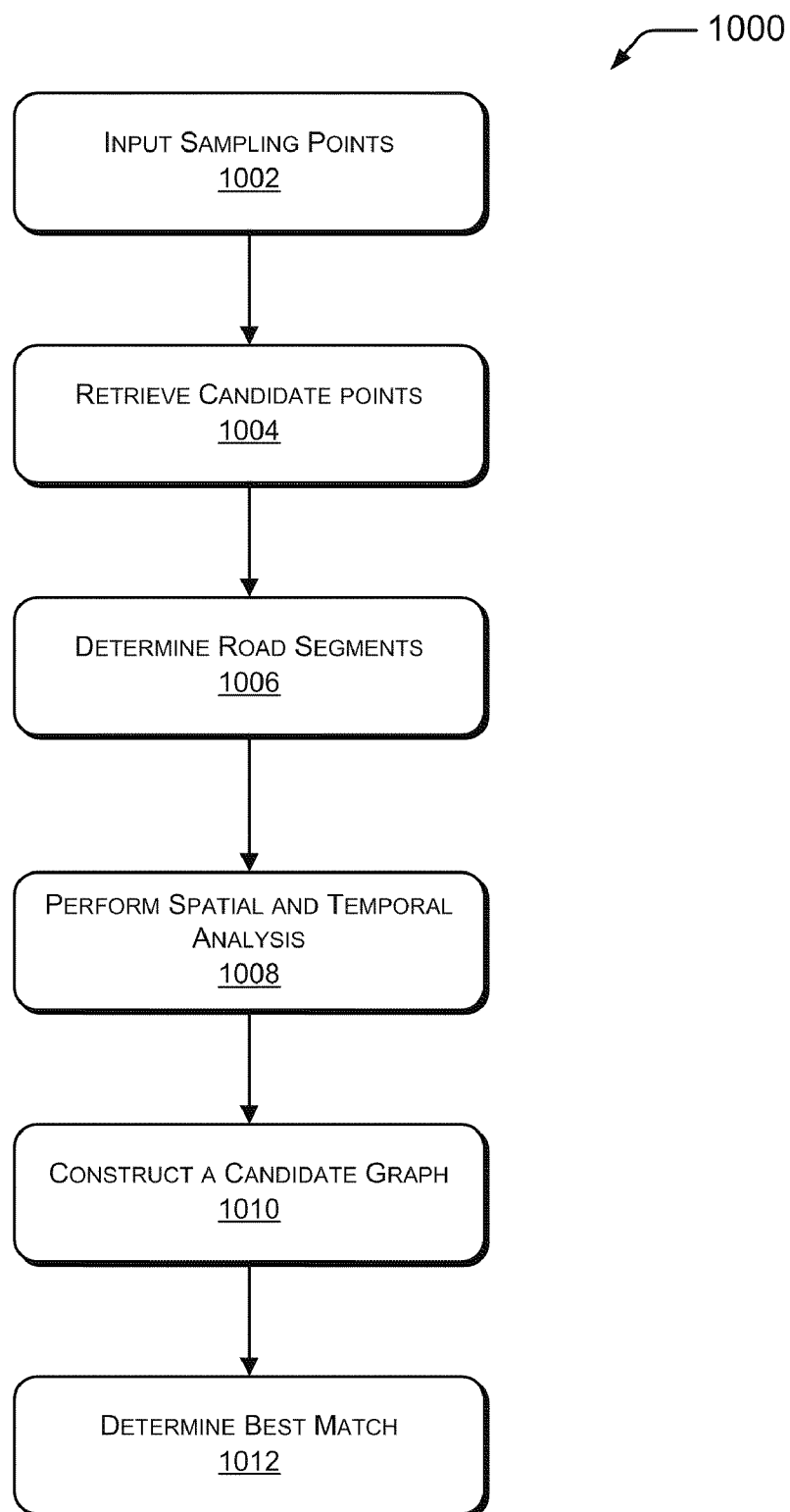


FIG. 10

$N:$

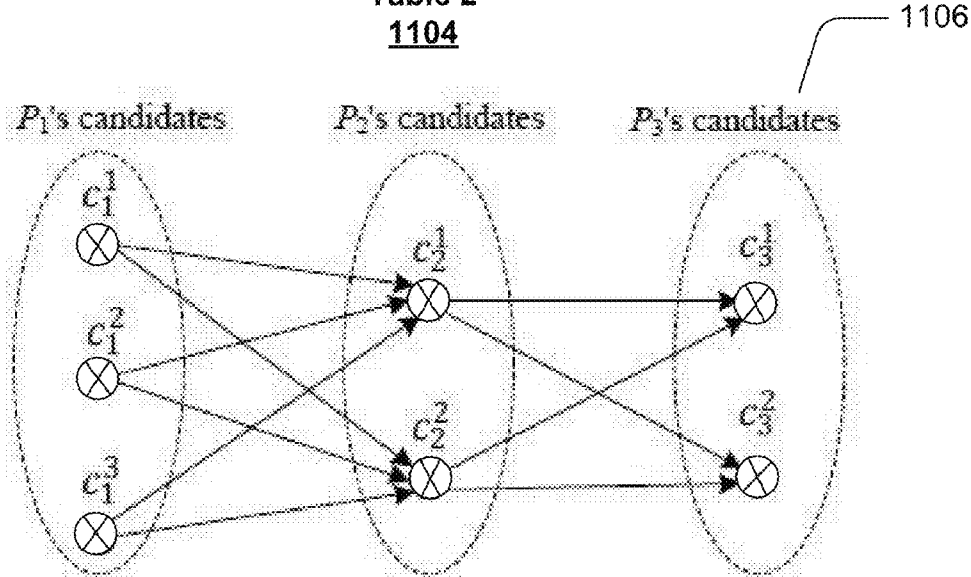
c_1^1	c_1^2	c_1^3	c_2^1	c_2^2	c_3^1	c_3^2
0.8	0.2	0.5	0.6	0.6	0.4	0.3

Table 1
1102

$(V, F_t):$

	$\rightarrow c_2^1$	$\rightarrow c_2^2$		$\rightarrow c_3^1$	$\rightarrow c_3^2$
c_1^1	(0.5,0.5)	(0.8,0.5)	c_2^1	(0.2,0.6)	(0.1,0.7)
c_1^2	(0.3,0.4)	(0.1,0.9)	c_2^2	(0.3,0.3)	(0.5,0.9)
c_1^3	(0.4,0.6)	(0.9,0.9)			

Table 2
1104



$f[\cdot]:$

c_1^1	c_1^2	c_1^3	c_2^1	c_2^2	c_3^1	c_3^2
0.8	0.2	0.5	0.95	1.04	1.076	1.175

Table 3
1108

FIG. 11

MAP-MATCHING FOR LOW-SAMPLING-RATE GPS TRAJECTORIES

BACKGROUND

[0001] There has been an increased use of handheld or dashboard-mounted travel guidance systems, for example, Global Positioning System (GPS)-embedded personal digital assistants (PDAs) and smart phones. In addition, there has been an increase in applications such as route planners, hot route finders, traffic flow analyzers, and geographical social network applications that use GPS data to achieve a better quality of service.

[0002] Typically, a GPS trajectory consists of a sequence of positions with latitude, longitude, instant speed, direction and timestamp information. However, this data can often be incorrect as a result of measurement errors caused by the limitations of typical GPS devices, as well as sampling errors caused by the sampling rate. Therefore, an observed GPS position often needs to be aligned with a road network on a digital map. This process is referred to as map-matching. The difficulty of map-matching can greatly differ depending on GPS accuracy and the sampling frequency, for example, map-matching is easier with data that is gathered frequently, and with a high degree of accuracy, than with data that is inaccurate or that is gathered less frequently.

[0003] Existing map-matching approaches generally employ an algorithm that maps sampled positions from a GPS trajectory onto vector road segments on a map. Such an approach typically considers sampled positions on a GPS trajectory while overlooking the speed and temporal data that may also be found in the GPS trajectory. These map-matching algorithms are typically most accurate when using data gathered at a high sampling rate. As sampling frequency decreases, measurement errors typically increase. However, while a high sampling rate results in increased accuracy, it also carries a greater computational cost.

[0004] Map-matching for low-sampling-rate GPS data is challenging because, as the sampling rate decreases, the interval between two neighboring positions in a trajectory increases, and less information is available to deduce the precise location of an object. A more effective approach for map-matching for low-sampling rate GPS trajectories utilizes temporal and speed data from the GPS trajectory to augment the spatial data.

SUMMARY

[0005] This summary is provided to introduce a selection of concepts in a simplified form that are further described below in the Detailed Description. This summary is not intended to identify key features or essential features of the claimed subject matter, nor is it intended to be used as an aid in determining the scope of the claimed subject matter.

[0006] In view of the above, this disclosure describes an exemplary method, user-interface, and computer-readable media for implementing map-matching for low-sampling rate GPS trajectories.

[0007] In an exemplary implementation, a mapping module receives a GPS trajectory that represents a path traveled by a user. For example, as the user is driving through town, a GPS device automatically records data at regular, predetermined time intervals. The GPS trajectory includes spatial data (e.g., one or more sampling points, latitude, longitude, and direction) and temporal data (e.g., speed and timestamp). A set of

one or more candidate projection points surrounding each of the one or more sampling points of the trajectory is retrieved from a road network database, and one or more candidate road segments upon which the candidate projection points lie are determined. Spatial and temporal analyses are performed on the retrieved set of one or more candidate projection points. A candidate graph is constructed based upon the results of the spatial analysis and the temporal analysis, and the candidate graph is evaluated to determine the set of candidate projection points that best matches the one or more sampling points received from the user.

[0008] Map-matching that is performed based on the GPS trajectory may also be used to calculate a recommended route based on user-submitted data that identifies additional locations to which the user would like to travel. For example, a user interface may be provided through which a user submits one or more locations to which the user would like to travel. Those points are appended to the existing GPS trajectory, and the mapping module calculates a route to the user-submitted locations based, at least in part, on the previously collected GPS trajectory. The calculated route is then presented to the user.

BRIEF DESCRIPTION OF THE DRAWINGS

[0009] The detailed description is described with reference to the accompanying figures. In the figures, the left-most digit of a reference number identifies the figure in which the reference number first appears. The use of the same reference numbers in different figures indicates similar or identical items.

[0010] FIG. 1 is a schematic of an illustrative architecture of a map-matching framework.

[0011] FIG. 2 is a block diagram of an exemplary computing device within the map-matching framework of FIG. 1.

[0012] FIG. 3 is a block diagram of an exemplary server within the map-matching framework of FIG. 1.

[0013] FIG. 4 is a flow chart of an exemplary map-matching process for determining a best-match trajectory.

[0014] FIG. 5 is an illustration of exemplary sampling points mapped to sets of candidate projection points along mapped road segments.

[0015] FIG. 6 is an illustrative shortest path computation.

[0016] FIG. 7 is a further illustration of the spatial-temporal analysis of FIG. 4.

[0017] FIG. 8 is an illustrative candidate graph within the map-matching framework of FIG. 1.

[0018] FIG. 9 is an illustrative local spatial-temporal matching strategy.

[0019] FIG. 10 is an illustrative process flow of the map-matching framework of FIG. 1.

[0020] FIG. 11 is an illustrative example of the map-matching framework of FIG. 1.

DETAILED DESCRIPTION

[0021] Map-matching for low-sampling rate global positioning system (GPS) trajectories is described. More specifically, an exemplary map-matching algorithm utilizes both a spatial analysis and a temporal analysis to analyze a submitted set of sampling points to determine a best match trajectory, or route, which is presented to the user.

[0022] FIG. 1 is a block diagram of an exemplary environment 100, which is used for map-matching for a low-sampling rate GPS on a computing device. The environment 100

includes an exemplary computing device **102**, which may take a variety of forms including, but not limited to, a portable handheld computing device (e.g., a personal digital assistant, a smart phone, a cellular phone), a laptop computer, a desktop computer, a media player, a digital camcorder, an audio recorder, a camera, or any other device capable of connecting to one or more network(s) **104** to log or to record daily activities for a user **106** (i.e., creating a location history). The computing device **102**, which connects to one or more network(s) **104**, is often associated with a user **106**. For example, the user **106** often carries their computing device **102** when travelling outside the home.

[0023] The network(s) **104** represent any type of communications network(s), including, but not limited to, wire-based networks (e.g., cable), wireless networks (e.g., cellular, satellite), cellular telecommunications network(s), and IP-based telecommunications network(s) (e.g., Voice over Internet Protocol networks). The network(s) **104** may also include traditional landline or a public switched telephone network (PSTN), or combinations of the foregoing (e.g., Unlicensed Mobile Access or UMA networks, circuit-switched telephone networks or IP-based packet-switch networks).

[0024] The computing device **102** accesses a global positioning system (GPS) that conveniently logs navigation and positioning information as the device moves with the user **106**. In an exemplary implementation, a GPS **108** in the computing device **102** starts recording location data upon detecting a satellite signal. For example, a GPS receiver collects the location data, p_1 , p_2 , and p_3 along a GPS trajectory **110** based on pre-determined rate. In the exemplary implementation, the GPS **108** in the computing device **102** uses a low sampling rate to minimize computations and network bandwidth usage. For example, the GPS **108** may record data every 2 minutes, every 5 minutes, or the like. The GPS **108** continues to collect the location data as long as there is a satellite signal detected on the network **104**.

[0025] The user **106** enters a starting location and a desired destination through a user interface of the computing device **102**. The user-submitted starting location and desired destination, along with the data collected by the GPS **108**, is sent over network **104** to servers **112**. Exemplary servers **112** include a map-matching module **114** that analyzes the data collected by the GPS to determine a preferred route, for example, the shortest and most direct route from the starting location to the user-submitted desired destination. The map-matching module **114** determines the preferred route by preparing candidate projection points, performing spatial and temporal analysis, and matching the results. The determined route is then stored and/or presented visually to the user. Alternatively, the data collected by the GPS may be used to determine a route consisting entirely of highways, a route consisting entirely of service roads, or the like. The map-matching module **114** provides the result back to the user via the computing device **102** through a network service provider, a context-aware computing service, email, text message, a pop up, and the like.

[0026] FIG. 2 illustrates an exemplary computing device **102**. The computing device **102** includes, without limitation, a processor **202**, a memory **204**, and one or more communication interfaces **206**. An operating system **208**, a user interface (UI) module **210**, a global positioning system (GPS) **108**, and content storage **212** are maintained in memory **204** and executed on processor **202**.

[0027] When executed on the processor **202**, the operating system **208** and UI module **210** collectively facilitate presentation of a user interface on a display of the computing device **102**. GPS **108** may be implemented as a component of a web browser or a search engine, or may be implemented as an application in the computing device **102**. As described above, the GPS **108** collects location data (e.g., GPS trajectories) over time as the computing device physically moves from one location to another. Content storage **212** provides local storage of sampling points and/or data received from map-matching module **114**. For example, the sampling points and/or data received from the map-matching module may be stored in GPS log **214**.

[0028] The communication interfaces **204** may include, without limitation, a wide area network (WAN) interface, a local area network interface (e.g., WiFi), a personal area network (e.g., Bluetooth) interface, and/or any other suitable communication interfaces to allow the computing device **102** to communicate over the network(s) **104**.

[0029] The computing device **102**, as described above, may be implemented in various types of system or networks. For example, the computing device may be a part of, without limitation, a client-server system, a peer-to-peer computer network, a distributed network, an enterprise architecture, a local area network, a wide area network, a virtual private network, a storage area network, and the like.

[0030] FIG. 3 illustrates an exemplary server within the map-matching framework of FIG. 1. The illustrated exemplary server **112** includes, without limitation, processor **302**, memory **304**, removable storage **306** and/or non-removable storage **308**, communication interface(s) **314**, and content storage **316**.

[0031] An operating system **310**, and a map-matching module **114** are maintained in the memory **304** and executed on the processor **302**. In an exemplary implementation, the map-matching module **114** includes a road network database **312** that includes, without limitation information pertaining to at least geographical locations within roadway system(s). For example, road network database **312** may contain a mapping system of the roadways of the greater Seattle area including, service roads, highways, and any other roads available to the user **106**. Map-matching module **114** may also include databases of other types of information, including for example, rivers, railways, airports, restaurants, hotels, etc.

[0032] Communication interfaces **314**, allow the processor **302** to communicate with the computing device **102**, other network servers, network storage, and/or other devices over the network(s) **104**, and content storage **316**. Content storage **316** may store the GPS data collected by the GPS **108** and sent to the server **112**. For example, the GPS data may be stored in GPS log **318**. Although not shown in FIG. 3, the server **112** may also include one or more known input device(s), such as a keyboard, a mouse, a pen, a voice input device, a touch input device, and an output device such as a display, speaker, printer, or the like.

[0033] Any memory described herein may include volatile memory (such as RAM), nonvolatile memory, removable memory, and/or non-removable memory, implemented in any method or technology for storage of information, such as, computer-readable instructions, data structures, applications, program modules, emails, and/or other content. Also, any of the processors described herein may include onboard memory in addition to or instead of the memory shown in the figures. The memory may include storage media such as, but

not limited to, random access memory (RAM), read only memory (ROM), flash memory, optical storage, magnetic disk storage or other storage devices, or any other medium which can be used to store the desired information and which can be accessed by the respective systems and devices.

[0034] FIG. 4 illustrates a process 400 for determining a best-match trajectory within the map-matching framework of FIG. 1. The map-matching framework of FIG. 1 enables the analysis of the data collected by the computing device to determine a preferred route from the starting location to the user desired destination.

[0035] The process 400 includes, without limitation, preparation of candidate projection points 402, spatial and temporal analyses 404, and result matching, storage, and visualization 406.

[0036] To prepare the candidate projection points, the map-matching module obtains data from GPS logs 408(1), 408(2), . . . , 408(N) and the road network database 312. A candidate computation 410 is then performed, the results of which are used to determine candidate sets 412. GPS logs 408 may correspond, for example, to GPS log 214 or GPS log 318 shown in FIG. 2 and FIG. 3, respectively.

[0037] Spatial and temporal analyses 404 are then performed. For example, a spatial analysis 414 and a temporal analysis 416 are performed on the candidate sets 412, resulting in a candidate graph 418.

[0038] The process concludes with result matching, storage, and visualization 406. For example, a best path search 420 is performed on the candidate graph 418 to determine a matching result 422. The matching result, or the preferred route, is presented to the user via a user interface 424.

[0039] Specific portions of process 400 are described below in greater detail with reference to FIG. 5-FIG. 9. However, it is to be appreciated that the description is not necessarily limited to the specific features or methods described.

[0040] FIG. 5 illustrates an exemplary candidate computation 410 portion of the process describe above with reference to FIG. 4. Referring back to FIG. 1 and FIG. 4, in an exemplary implementation, as the user 106 travels along the trajectory 110, raw GPS trajectory data, or sampling points, are automatically collected by computing device 102 and communicated to the map-matching module 114. The sampling points are taken at a low frequency, such as every 2 to 5 minutes, and are stored in GPS logs 408. As described above, GPS logs 408 may correspond to the GPS log 214 in the computing device 102 and/or the GPS log 318 in the server 112. Map-matching module 114 accesses the sampling points stored in the GPS log 408 to determine the best trajectory or route to get the user 106 from an identified starting location to a desired destination by first determining a set of candidate projection points corresponding to the sampling points stored in the GPS log 408. To determine the set of candidate projection points, map-matching module 114 utilizes a road network database 312 which, as discussed above, includes possible candidate road segments, or routes, available to the user 106. Using the information from the road network database 312 and GPS log 408, map-matching module 114 performs a candidate computation 404, an example of which is shown in FIG. 5.

[0041] Exemplary candidate computation 410 is determined by plotting one or more sampling points, p_i , along a determined candidate road segment. For each sampling point p_i , a circle 502 encompasses a set of candidate road segments supplied by road network database 312 within a given radius. In the example illustrated in FIG. 5, three sampling points are shown (i.e., p_{i-1} , p_i , and p_{i+1}). Focusing first on sampling point p_{i-1} , two candidate road segments (i.e., r_{i-1}^1 and r_{i-1}^2)

lie within the circle 502(1) of p_{i-1} , indicating that the sampled point p_{i+1} may correspond to a point on either road segment r_{i-1}^1 or road segment r_{i-1}^2 . Utilizing this data, the map-matching module 114 determines at least one set of candidate projection points that lie along the candidate road segments r_{i-1}^1 and r_{i-1}^2 . For example, as further shown in FIG. 5, the map-matching module 114 determines candidate projection points c_{i-1}^1 and c_{i-1}^2 , corresponding to sampling point p_{i-1} .

[0042] Similarly, the map-matching module 114 also identifies candidate projection points c_i^1 and c_i^2 along candidate road segments r_i^1 and r_i^2 , respectively, within the circle 502(2) encompassing sampling point p_i ; and identifies candidate projection points c_{i+1}^1 and c_{i+1}^2 corresponding to sampling point p_{i+1} , where sampling point p_{i+1} is encompassed by circle 502(3).

[0043] The result of the candidate computation 410 is, for each sampling point, a set of candidate projection points. Referring to the example shown in FIG. 5, the candidate computation 410 results in set $\{c_{i-1}^1, c_{i-1}^2\}$, corresponding to sampling point p_{i-1} ; set $\{c_i^1, c_i^2\}$ corresponding to sampling point p_i ; and set $\{c_{i+1}^1, c_{i+1}^2\}$ corresponding to sampling point p_{i+1} .

[0044] In an exemplary implementation, one or more indexing techniques may be used to expedite the generation of a set of candidate projection points. For example, indexing techniques that may be used include, without limitation, a space-partition based indexing method, such as the grid-based spatial index and the quad tree indexing structure, or a data driven indexing structure such as an R-tree indexing structure.

[0045] After determining the set of candidate projection points, as described above, the map-matching algorithm performs a spatial analysis 414 and a temporal analysis 416 to identify a particular candidate projection point within each set of candidate projection points that best matches the corresponding sampling point.

[0046] Exemplary spatial analysis 414 utilizes geometric and topological information from the road network database 312 to evaluate each of the determined candidate projection points. In one implementation, spatial analysis 414 includes two components, an observation probability component and a transmission probability component.

[0047] The observation probability component represents the likelihood that a trajectory, or route, supplied from the map-matching module 114 to the user 106 is the best route based upon the distance between two candidate points, such as candidate points c_i^1 and c_i^2 . However, because no additional information is taken into account the result typically includes an error.

[0048] The error in the observation probability is calculated as a normal distribution $N(\mu, \sigma^2)$ using the distance between sampling point p_i and candidate point c_i^j . The normal distribution demonstrates how likely the trajectory supplied from the map-matching module 114 would have been if the user 106 had actually been on a road with a location c_i^j and without considering any previous points. Accordingly, the observation probability is calculated according to:

$$N(c_i^j) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x_i^j - \mu)^2}{2\sigma^2}} \quad \text{Equation (1)}$$

[0049] where x is the distance between the sampling point p_i and its corresponding candidate point c_i^j , represented as $x_i^j = \text{dist}(c_i^j, p_i)$. In one implementation, a zero-mean normal distribution with a standard deviation of about 66 feet (or

about 20 meters) may be used. Alternatively, any normal distribution with a suitable standard deviation may be used.

[0050] It is assumed that a typical user **106** would desire the shortest and most direct route to a destination. Therefore, in one implementation, Equation (1) corresponds to a shortest path computation. A variety of shortest path algorithms may be used to compute the shortest path.

[0051] FIG. 6 illustrates an exemplary shortest path computation utilizing a Dijkstra's shortest path algorithm. Steps **602**, **604**, **606**, **608**, **610**, and **612**, illustrate typical steps of a Dijkstra's computation. An example Dijkstra's algorithm is:

```

1. INITIALIZE SINGLE-SOURCE (G, s)
2. S ← { } // S will ultimately contains vertices of final shortest-path
   weights from s
3. Initialize priority queue Q i.e., Q ← V[G]
4. while priority queue Q is not empty do
5.     u ← EXTRACT_MIN(Q) // Pull out new vertex
6.     S ← S ∪ {u}
       // Perform relaxation for each vertex v adjacent to u
7.     for each vertex v in Adj[u] do
8.         Relax (u,v,w)

```

where G represents a candidate graph, s represents a source vertex or source sampling point, Q is a set of candidate projection points, and a vertex represents sampling points other than the source sampling point. A tree T is built that spans all reachable vertices from a point S. Vertices are added to the tree T in order of the distance between the source sampling point and the remaining sampling points. For example, a first distance S, then a vertex closest to S, and so on.

[0052] As described above, an exemplary spatial analysis **414** has an observation probability component and a transmission probability component. The example Dijkstra's algorithm described above results in an observation probability.

[0053] The transmission probability for candidate points c_{i-1}^t to c_i^s for two neighboring sampling points p_{i-1} and p_i respectively, is the likelihood that the "true" path from p_{i-1} and p_i follows the shortest path from c_{i-1}^t to c_i^s . Such a true path is represented by:

$$V(c_{i-1}^t \rightarrow c_i^s) = \frac{d_{i-1 \rightarrow i}}{w_{(i-1,t) \rightarrow (i,s)}} \quad \text{Equation (2)}$$

where $d_{i-1 \rightarrow i} = \text{dist}(p_i, p_{i-1})$ is the Euclidean distance between p_i and p_{i-1} , and $w_{(i-1,t) \rightarrow (i,s)}$ is the length of the shortest path from c_{i-1}^t to c_i^s .

[0054] The product of the observation probability and the transmission probability reflect the likelihood that the user **106** will move from c_{i-1}^t to c_i^s . Combining the observation probability and the transmission probability, an exemplary spatial analysis function is:

$$F_s(c_{i-1}^t \rightarrow c_i^s) = N(c_i^s) * V(c_{i-1}^t \rightarrow c_i^s), 2 \leq i \leq n \quad \text{Equation (3)}$$

where c_{i-1}^t and c_i^s may be any two candidate points for two neighboring sampling points p_{i-1} and p_i . The spatial analysis **414** for a candidate p_i will differ depending upon the previous candidate point.

[0055] As previously discussed, map-matching based on spatial analysis alone, typically leads to poor accuracy when determining a best trajectory or route. For example, as illustrated in FIG. 7, a bold line **702** represents a highway, and a thin line **704** represents a service road. Just looking at sam-

pling point p_i , it is difficult to determine from which road sampling point p_i was most likely sampled. In this example, although p_i appears closer to the service road **704**, the neighboring point's p_{i-1} and p_{i+1} are closer to the highway, which may suggest that p_i was more likely sampled from a location on the highway than a location on the service road. This example illustrates the potential for errors when performing map-matching based solely on spatial analysis.

[0056] Analyzing speed information in addition to the location information can increase the accuracy of the map-matching. In the example shown in FIG. 7, because the service road **704** and highway **702** are located in close proximity to one another, whether p_i is on the highway or the service road, the candidate points c_i^1 and c_i^2 have similar spatial measurements. However, if the average speed of the user **106** is calculated over a time interval Δt , a speed indicating a highway or a service road may be determined, enabling a more accurate result when determining the best trajectory or route for the user. For example, if a typical service road has a speed limit of 30 miles per hour (mph), and over the time interval, an average speed of 65 (mph) is calculated, then it would be more reasonable to map sampling point p_i to the candidate projection point c_i^2 on the highway.

[0057] Therefore, an exemplary temporal analysis **416** is based upon an average speed between two candidate points c_{i-1}^t and c_i^s corresponding to two neighboring sampling points, p_{i-1} and p_i , respectively. The average speed $v_{(i-1,t) \rightarrow (i,s)}$ is calculated using the formula:

$$v_{(i-1,t) \rightarrow (i,s)} = \frac{\sum_{u=1}^k l_u}{\Delta t_{i-1 \rightarrow i}} \quad \text{Equation (4)}$$

where $l_u = e_u^t * 1$ is the length of road segment e_u^t , and $\Delta t_{i-1 \rightarrow i} = p_i^t - p_{i-1}^t$ is the time interval between the two sampling points p_i and p_{i-1} . In an exemplary implementation, each road segment e_u^t is also associated with a typical speed value $e_u^t * v$, and a cosine distance may be used to measure the similarity between the actual average speed from c_{i-1}^t to c_i^s and the speed constraints (e.g., known speed limits) of the path. Alternatively, any suitable measurement may be used to determine the similarity. Considering a vector k with elements of the same value $v_{(i-1,t) \rightarrow (i,s)}$ and the vector $(e_1^t v, e_2^t v, \dots, e_k^t v)^T$, an exemplary temporal analysis function is calculated as:

$$F_t(c_{i-1}^t \rightarrow c_i^s) = \frac{\sum_{u=1}^k (e_u^t \cdot v \times v_{(i-1,t) \rightarrow (i,s)})}{\sqrt{\sum_{u=1}^k (e_u^t \cdot v)^2 \times \sum_{u=1}^k v_{(i-1,t) \rightarrow (i,s)}^2}} \quad \text{Equation (5)}$$

[0058] Combining Equation (3) and Equation (5), as set forth above, the spatial-temporal (ST) function for $c_{i-1}^t \rightarrow c_i^s$ is:

$$F(c_{i-1}^t \rightarrow c_i^s) = F_s(c_{i-1}^t \rightarrow c_i^s) * F_t(c_{i-1}^t \rightarrow c_i^s), 2 \leq i \leq n \quad \text{Equation (6)}$$

[0059] Referring back to the process illustrated in FIG. 4, a candidate graph **418** is created using candidate projection

points determined during the spatial analysis **414** and the temporal analysis **416** calculations. The candidate graph **418** illustrates the possible trajectories or routes available to the user **106** and is used to perform a best path search **420**.

[0060] FIG. **8** illustrates an exemplary best path search **420**. For example, FIG. **8** illustrates an exemplary candidate graph $G'_T(P'_T, E'_T)$ for a trajectory $T: p_1 \rightarrow p_2 \rightarrow \dots \rightarrow p_n$, where V'_T is a set of candidate points corresponding to each of a set of received sampling points, and E'_T is a set of edges representing the shortest paths between any two neighboring candidate points.

[0061] The candidate path search **420** may include $c_1^{s1} \rightarrow c_2^{s2} \rightarrow \dots \rightarrow c_n^{sn}$, for example, **802** \rightarrow **804** \rightarrow **806**.

[0062] To determine which candidate path sequence is a best match trajectory or route **422** corresponding to the sample points input by the user **106** or collected by the GPS **108**, a score for each of the candidate path sequences is calculated. For example, scores for candidate path sequences **808(1)** and **808(2)** may be calculated and compared to determine which of the two candidate path sequences has the highest score, and is therefore the best match for the sampling points along the trajectory. In an exemplary implementation, the score for such a candidate sequence path is given by:

$$F(P_c) = \sum_{i=2}^n F(c_{i-1}^{s_{i-1}} \rightarrow c_i^{s_i}) \quad \text{Equation (7)}$$

[0063] The best-match path P for a trajectory T may be selected using:

$$P = \arg \max_{P_c} F(P_c), \forall P_c \in G'_T(V'_T, E'_T) \quad \text{Equation (8)}$$

[0064] Algorithm 1, set forth below, outlines an exemplary algorithm for a spatial and temporal analysis **404** using the map-matching module **114**. Algorithm 1 includes the terms and equations described above in FIGS. **1-7** and Equations **1-6**. According to Algorithm 1, a set of candidate points for each sampling point along a Trajectory T (for example, the sampling points found in the GPS logs **408**) is calculated. A candidate graph is constructed based upon the spatial and temporal analyses, followed by a report identifying the path sequence P with the highest ST-function value from G'_T . The result is sent to Algorithm 2.

Algorithm 1 Map-Matching Algorithm

Input: Road Network G , a list of GPS points p_1, p_2, \dots, p_n
Output: The matched sequence $c_{1,j1}, c_{2,j2}, \dots, c_{n,jn}$
1: Initialize tList as an empty list; // a list of sets of candidates
2: for $I = 1$ to n do
3: $s = \text{GetCandidates}(p_i, G, r)$; // candidates within radius r
4: tList.add(s);
5: $G' = \text{ConstructGraph}(\text{tList})$; // constructs graph G'
6: RETURN FindMatchedSequence (G')

[0065] Algorithm 2, set forth below, outlines an exemplary algorithm for result matching, storage and visualization **406** based upon the results of Algorithm 1. Algorithm 2 determines the trajectory or route most likely to correspond to the received sampling points.

Algorithm 2 FindMatchedSequence

Input: Candidate graph G'
Output: The matched sequence $c_{1,j1}, c_{2,j2}, \dots, c_{n,jn}$
1: Let $f[]$ denote the highest score found so far;
2: Let $\text{pre}[]$ denote the parent of current candidate;
3: for each $c_{i,s}$ do
4: $f[c_{i,s}] = F_s(c_{i,s})$;

-continued

Algorithm 2 FindMatchedSequence

5: for $i = 2$ to n do
6: for each $c_{i,s}$ do
7: $\text{max} = -\infty$;
8: for each $c_{i-1,t}$ do
9: $\text{alt} = f[c_{i-1,t}] + F_s(c_{i,s}) * F_t(P_{c_{i-1,t} \text{ to } c_{i,s}})$;
10: if ($\text{alt} > \text{max}$) then
11: $\text{max} = \text{alt}$;
12: $\text{pre}[c_{i,s}] = c_{i-1,t}$;
13: $f[c_{i,s}] = \text{max}$;
14: Initialize rList as an empty list;
15: $p = \text{maxarg}_{c_{n,s}}(f[c_{n,s}])$;
16: for $I = n$ down to 1 do
17: rList.add(p);
18: $p = \text{pre}[p]$;
19: return rList.reverse90;

[0066] The exemplary ST-matching strategy represented by Algorithm 1 and Algorithm 2 is based upon a global algorithm, meaning the best path sequence is computed based upon an overall score for the entire trajectory of the candidate path sequence. However, if a trajectory has too many sampling points (i.e., n is very large) map-matching module **114** may use a local algorithm, based on a subset of the sampling points, to determine a matching result **422**.

[0067] FIG. **9** illustrates an exemplary best path search **420** using a local algorithm. As illustrated in FIG. **9**, each partial candidate graph **902** and **904** is constructed from a trajectory T . A best matching sequence is determined for each partial candidate graph similar to the global algorithm approach set forth above. The process is repeated for each partial candidate graph to determine a matching result **422**.

[0068] FIG. **10** illustrates an exemplary method outlining the map-matching procedure set forth above. At block **1002**, a set of sampling points are collected by GPS **108** on computing device **102**. The sampling points are communicated from the computing device **102** to the map-matching module **114** over network **104**. At block **1004** map-matching module **114** generates a set of candidate projection points corresponding to each sampling point. For example, for every p_1 , a set of candidate points may include c_1^1 , c_1^2 , and c_1^3 . Using these candidate points, map-matching module **114** may access a road network database at block **1006** to determine one or more corresponding road segments. At block **1008**, map-matching module **114** may perform a spatial analysis and a temporal analysis using the set of candidate points. For example, an observation measurement, a transmission measurement, and a temporal measurement are illustrated in Tables 1 and 2, respectively, shown in FIG. **11**.

[0069] At block **1010** a candidate graph may be constructed. For example, a candidate graph **1106**, may be created corresponding to those measurements set forth in Tables 1 and 2, **1102** and **1104** of FIG. **11**. At block **1012** map-matching module may determine a best match trajectory for the sampling points input by GPS **108**. For example, map-matching module **114** may use Algorithms 1 and 2, described above, to ascertain which candidate point has the highest overall score, and therefore is the best match to the sampling points. For example, Table 3, **1108**, illustrated in FIG. **11**, shows c_3^2 having the highest overall score for candidate graph **1106**. Therefore, c_3^2 may be chosen as the matching result for p_3 , and therefore, the best match trajectory is most likely

$$c_1^1 \rightarrow c_2^2 \rightarrow c_3^2.$$

CONCLUSION

[0070] Although map-matching for low-sampling rate GPS trajectories has been described in language specific to structural features and/or methods, it is to be understood that the subject of the appended claims are not necessarily limited to the specific features or methods described. Rather, the specific features and methods are disclosed as exemplary implementations.

What is claimed is:

1. A computer-implemented method comprising:
 - implementing a processor to execute computer executable instructions stored on a computer-readable medium to perform operations comprising:
 - receiving a global positioning system (GPS) trajectory comprising a plurality of sampling points;
 - determining a set of one or more candidate projection points for each sampling point of the plurality of sampling points;
 - performing a spatial analysis and a temporal analysis on the retrieved sets of one or more candidate points;
 - constructing a candidate graph based upon the results of the spatial analysis and the temporal analysis; and
 - evaluating the candidate graph to determine a best-match trajectory that corresponds to the plurality of sampling points.
2. The method of claim 1, wherein performing the spatial analysis comprises:
 - determining an observation probability; and
 - determining a transmission probability.
3. The method of claim 2, wherein the observation probability is calculated according to the distance between a sampling point and a candidate projection point.
4. The method of claim 2, wherein the transmission probability is calculated according to a plurality of candidate projection points and the corresponding plurality of sampling points, such that the distance between the plurality of sampling points follows the shortest path from a first candidate projection point to a second candidate projection point.
5. The method of claim 1 further comprising presenting the best-match trajectory on a display.
6. The method of claim 1, wherein the determining is performed by a built-in grid-based spatial index method.
7. The method of claim 1, wherein the evaluating the candidate graph is based upon an overall score for a candidate sequence path, wherein the overall score indicates the probability that the candidate sequence path is the best-match trajectory.
8. The method of claim 1, wherein the plurality of sampling points comprises data gathered by a GPS based upon a pre-determined, low-frequency sampling interval.
9. The method of claim 8, wherein the low-frequency sampling interval comprises every 2 minutes, every 3 minutes, every 4 minutes, every 5 minutes, or a time period greater than every 5 minutes.

10. One or more computing devices, comprising:
 - one or more processors;
 - a memory coupled to the one or more processors; and
 - a map-matching module stored in the memory and executed on the processor to:
 - receive a plurality of sampling points collected along a user-travelled trajectory;
 - determine a best-match trajectory corresponding to the plurality of sampling points based upon a spatial-temporal analysis; and
 - present the best-match trajectory.

11. The one or more computing devices of claim 10, wherein the sampling points are collected at a pre-determined, low-frequency sampling interval.

12. The method of claim 11, wherein the low-frequency sampling interval comprises every 2 minutes, every 3 minutes, every 4 minutes, every 5 minutes, or a time period greater than every 5 minutes.

13. The one or more computing devices of claim 10, wherein a candidate path sequence is a result of the spatial-temporal analysis and is the best match trajectory corresponding to the one or more sampling points.

14. The one or more computing devices of claim 12, wherein the pre-determined sampling interval is defined by the user.

15. The one or more computing devices of claim 10, wherein the best match trajectory is further based upon a candidate set comprising a plurality of candidate projection points

16. The one or more computing devices of claim 15, wherein the spatial-temporal analysis employs a road network database comprising geometric information and topological information to evaluate the plurality of candidate projection points.

17. One or more computer-readable media storing computer-executable instructions that, when executed on one or more processors, perform operations comprising:

- performing a spatial-temporal analysis on a plurality of candidate projection points, the plurality of candidate projection points correspond to a plurality of sampling points collected along a user-travelled trajectory;
- constructing a candidate graph based upon the plurality of candidate projection points;
- evaluating the candidate graph to determine a best-match trajectory; and
- presenting the best-match trajectory.

18. The one or more computer-readable media of claim 17, wherein the plurality of sampling points are gathered during a pre-defined low-frequency sampling interval.

19. The one or more computer-readable media of claim 17, wherein the candidate graph comprises one or more partial candidate graphs.

20. The one or more computer-readable media of claim 17, wherein the best-match trajectory is determined based on a score calculated for each candidate sequence path on the candidate graph.

* * * * *