METHOD AND DEVICE FOR CORRECTING ERRORS IN MAP DATA

TECHNICAL FIELD

[0001] Various aspects of this disclosure relate to methods and devices for correcting errors in map data.

BACKGROUND

[0002] The quality of an e-hailing server which enables customers to hail taxis using their smartphones largely depends on the underlying map data which is for example used for estimating the time when the driver will be there to pick up the user, a price of the ride and how long it will take to get to the destination. Accordingly, it is desirable to keep this map data up-to-date with the real world. However, updating of maps may require a high amount of resource in view of the high number of ever-changing maps. Moreover, manual updating of maps has the risk of introducing human errors. Accordingly, approaches for making updating and enhancing of map data more efficient and less error-prone are desirable.

SUMMARY

[0003] Various embodiments concern a method for correcting errors in map data including obtaining map data specifying travel possibilities between locations, generating a routing graph from the map data by assigning a vertex to each location and an edge from one vertex to another vertex if the map data specifies that it is possible to travel from the location to which the first vertex is assigned to the location to which the second vertex is assigned (e.g. if the map data specifies that it is possible to travel directly along a road segment from the location to which the first vertex is assigned to the location to which the second vertex is assigned), identifying the largest strongly connected component of the routing graph, identifying one or more further strongly connected components of the routing graph, detecting errors in the map data by identifying travel possibilities that are not in the map data which lead to missing connections between the one or more further strongly connected components of the routing graph and the largest strongly connected component of the routing graph and amending the map data by the identified travel possibilities.

[0004] According to one embodiment, the travel possibilities are directed travel possibilities and the routing graph is a directed routing graph.

[0005] According to one embodiment, the travel possibilities include roads associated with an allowed travel direction.

[0006] According to one embodiment, the method includes associating each further strongly connected component of the routing graph with a connectivity class of a plurality of connectivity classes, wherein the connectivity classes include one or more of a class of strongly connected components each being isolated from the largest strongly connected component and including multiple vertices, a class of strongly connected components each having a single vertex isolated from the other strongly connected components, a class of strongly connected components each having a single vertex which can only be reached from a vertex of one or more other strongly connected components, and a class of strongly connected components each having a single vertex which cannot be reached from any other strongly connected component but from which strongly connected component can be reached, wherein the method further includes detecting errors in the map data using the connectivity classes associated with the further strongly connected components of the routing graph.

[0007] According to one embodiment, the method includes associating groups of two or more strongly connected components of the routing graph with a connectivity relation of a plurality of connectivity relationships, wherein the connectivity relationships include one or more of a connectivity relationship indicating that there are edges between the strongly connected components of the group, a connectivity relationship indicating that the group of strongly connected components forms a maximum path of single vertices and a connectivity relationship indicating that historic driving data of a geographical region represented by the map data indicates that a driver has travelled between vertices of the strongly connected components which are not connected in the routing graph, wherein the method further includes detecting errors in the map data using the connectivity relationships associated with the groups of strongly connected components of the routing graph.

[0008] According to one embodiment, the method further includes acquiring historic driving data of a geographical region represented by the map data, determining whether there exists a pair of vertices including a first vertex and a second vertex in the routing graph wherein the map data does not specify a travel possibility form the first vertex to the second vertex but the historic driving data indicates that a driver has travelled from the location to which the first

vertex is assigned to the location to which the second vertex is assigned and determining that the map data lacks a travel possibility from the location to which the first vertex is assigned to the location to which the second vertex is assigned if such a pair of vertices exists.

[0009] According to one embodiment, the method includes iteratively amending the map data by iteratively repeating the generating of a routing graph, identifying the largest strongly connected component of the routing graph, identifying one or more further strongly connected components of the routing graph, detecting errors and amending the map data, wherein the map data amended in a previous iteration is used as map data for a current iteration.

[0010] According to one embodiment, the method includes monitoring a quality of the map data by iteratively repeating the obtaining of map data, generating of a routing graph, identifying the largest strongly connected component of the routing graph and identifying one or more further strongly connected components of the routing graph and determining a quality indicator of the routing graph and tracking the quality indicator over time.

[0011] According to one embodiment, the method includes generating an alarm signal when the quality indicator falls below a predetermined threshold.

[0012] According to one embodiment, the method includes exporting a specification of the identified strongly connected components of the routing graph to a visualization tool.

[0013] According to one embodiment, the method further includes visualizing the identified strongly connected components of the routing graph.

[0014] According to one embodiment, the visualizing further includes visualizing connectivity classes of strongly connected components or connectivity relationships of groups of two or more strongly connected components or both.

[0015] According to one embodiment, the visualizing further includes overlaying the strongly connected components with a geographic map given by the map data.

[0016] According to one embodiment, the method further includes navigating one or more vehicles by the amended map data.

[0017] According to one embodiment, the method further includes transmitting the amended map data to a vehicle and controlling the vehicle using the amended map data.

[0018] According to various embodiments, a server computer including a communication interface, a memory and a processing unit configured to perform the method of any one of the above embodiments is provided.

[0019] According to one embodiment a computer program element is provided including program instructions, which, when executed by one or more processors, cause the one or more processors to perform the method of any one of the above embodiments.

[0020] According to one embodiment a computer-readable medium is provided including program instructions, which, when executed by one or more processors, cause the one or more processors to perform the method of any one of the above embodiments.

BRIEF DESCRIPTION OF THE DRAWINGS

[0021] The invention will be better understood with reference to the detailed description when considered in conjunction with the non-limiting examples and the accompanying drawings, in which:

- FIG. 1 shows a communication arrangement including a smartphone and a server.
- FIG. 2 illustrates an error of map data.
- FIG. 3 shows an example of a routing graph and a strongly connected component (SCC) of the routing graph.
- FIG. 4 shows a flow diagram according to an embodiment illustrating a processing of map data according to an embodiment.
- FIG. 5 shows an example visualization of the result of the processing of map data.
- FIG. 6 shows an example where a poor or outdated satellite image leads to map data which gives rise to SCCs which are disconnected from the largest SCC in the routing graph.
- FIG. 7 shows an example wherein the map data misses a travel possibility between two lanes in two cases.
- FIG. 8 shows a flow diagram illustrating a method for correcting errors in map data according to an embodiment.
- FIG. 9 shows a server computer according to an embodiment.

DETAILED DESCRIPTION

[0022] The following detailed description refers to the accompanying drawings that show, by way of illustration, specific details and embodiments in which the disclosure may be practiced. These embodiments are described in sufficient detail to enable those skilled in the art to practice the disclosure. Other embodiments may be utilized and structural, and logical

changes may be made without departing from the scope of the disclosure. The various embodiments are not necessarily mutually exclusive, as some embodiments can be combined with one or more other embodiments to form new embodiments.

[0023] Embodiments described in the context of one of the devices or methods are analogously valid for the other devices or methods. Similarly, embodiments described in the context of a device are analogously valid for a vehicle or a method, and vice-versa.

[0024] Features that are described in the context of an embodiment may correspondingly be applicable to the same or similar features in the other embodiments. Features that are described in the context of an embodiment may correspondingly be applicable to the other embodiments, even if not explicitly described in these other embodiments. Furthermore, additions and/or combinations and/or alternatives as described for a feature in the context of an embodiment may correspondingly be applicable to the same or similar feature in the other embodiments.

[0025] In the context of various embodiments, the articles “a”, “an” and “the” as used with regard to a feature or element include a reference to one or more of the features or elements.

[0026] As used herein, the term “and/or” includes any and all combinations of one or more of the associated listed items.

[0027] In the following, embodiments will be described in detail.

[0028] An e-hailing app, typically used on a smartphone, allows its user to hail a taxi (or also a private driver) through his or her smartphone for a trip.

[0029] FIG. 1 shows a communication arrangement including a smartphone 100 and a server (computer) 106.

[0030] The smartphone 100 has a screen showing the graphical user interface (GUI) of an e-hailing app that the smartphone’s user has previously installed on his smartphone and has opened (i.e. started) to e-hail a ride (taxi or private driver).

[0031] The GUI 101 includes a map 102 of the user’s vicinity (which the app may determine based on a location service, e.g. a GPS-based location service). Further, the GUI 101 includes a box for point of departure 103 (which may be set to the user’s present location obtained from location service) and a box for destination 104 which the user may touch to enter a destination (e.g. opening a list of possible destinations). There may also be a menu (not shown) allowing the user to select various options, e.g. how to pay (cash, credit card, credit balance of the e-hailing service). When the user has selected a destination and made any necessary option selections, he or she may touch a “find car” button 105 to initiate searching of a suitable car.

[0032] For this, the e-hailing app communicates with the server 106 of the e-hailing service via a radio connection. The server 106 includes a database 107 knowing the current location of registered drivers, when they are expected to be free, has information about traffic jams etc. From this, a processor 110 of the server 106 determines the most suitable driver (if available) and provides an estimate of the time when the driver will be there to pick up the user, a price of the ride and how long it will take to get to the destination. The server communicates this back to the smartphone 100 and the smartphone 100 displays this information on the GUI 101. The user may then accept (i.e. book) by touching a corresponding button.

[0033] To determine all route-related information, like the most suitable driver and an estimate of the time when the driver will be there to pick up the user, a price of the ride and how long it will take to get to the destination, the processor 110 access a database 107 storing map data including for example one or more road maps indicating where roads are located, the allowed direction of travel, speed limits, etc.

[0034] The database 107 is in this example implemented by a local memory 109 of the server computer 106. However, it may also be implemented at least partially externally to the server computer 106, e.g. in a cloud, and it may be filled by access to an external database 111, e.g. an open route map information database such as OSM (Open Street Map).

[0035] For determination of information like described above the map data 108 should be up to date and free of errors, i.e. should not lack travel possibilities (e.g. roads or allowed directions of travel) that exist in reality (i.e. in the physical geographic area represented by the map data). However, errors are typically introduced, for example by human error or inaccurate measurements (e.g. inaccurate (e.g. outdated) satellite images, positioning inaccuracies when driving along a route etc.).

[0036] FIG. 2 illustrates an error of map data where the direction of (allowed) travel on a first part of a road 201 is different from a second part of the road 202 such that the direction of travel changes at a location 203. It is typically practically impossible to identify errors of this kind unless there is a complaint specifically regarding this case (e.g. by a driver expected to travel along the road 202 in the wrong direction).

[0037] Similar issues can occur due to wrong road restrictions and turn restrictions.

[0038] According to various embodiments, an approach which allows detecting problems in map data, e.g. of the kind as described above, is provided.

[0039] This approach makes, according to various embodiments, use of the fact that a routing graph should be strongly connected. This means that the graph which represents locations (represented by vertices of the graph) connected by roads (represented by roads of the graph) should ideally allow routing between any two points (vertices) in the graph. This may especially be an issue for map data of developing countries: in cities of such countries various map quality issues can cause this property not to hold. Various embodiments allow bringing these discrepancies to light.

[0040] Specifically, according to various embodiments, an approach is provided that (e.g. automatically) surfaces issues related to map connectivity and consists of a full end to end pipeline for continued improvement at low operational cost. It helps in finding, among other issues:

- Missing roads
- Incorrect directions of travel
- Incorrect road restrictions
- Incorrect turn restrictions
- Isolated or disconnected roads.

[0041] According to one embodiment, a routing graph (e.g. generated by the processor 110 from the map data 108) is decomposed into different elements which can be meaningfully classified and related to one another in terms of their connectivity. This classification system is integrated in a closed loop workflow that results in continued (iterative) improvements of the map data 108.

[0042] Thus, according to various embodiments, a closed loop is provided that utilizes both automation and manual operations work in a seamless way. This allows improving maps especially in regions with low quality and unplanned, quickly changing, road structure.

[0043] According to various embodiments, the processing of map data 108 includes connectivity classification and a pipeline that creates a feedback loop for map improvements.

[0044] Connectivity classification finds and classifies different parts of a routing graph generated from map data 108. This includes a classification of subgraphs of the routing graph based on their relationship with the full routing graph, and an algorithm for finding these classes in a scalable way.

[0045] Connectivity relationships are defined for groups (e.g. pairs) of different strongly connected components (SCC) of the whole routing graph. Connectivity classes are defined for

individual SCCs. First the graph is broken down into its different SCCs and then each SCC is classified into one of multiple SCC classes based on its likely effect on the overall connectivity of the graph.

[0046] A strongly connected component is a subgraph of a graph in which there is a path between all pairs of vertices of the subgraph.

[0047] FIG. 3 shows an example of a graph having vertices A, B, C, D, E, F, G wherein the subgraph including vertices B, C, D, E forms a strongly connected component.

[0048] To determine the SCC class of an SCC and connectivity relationships between SCCs, the processor 110, for example, first identifies the SCC components of a routing graph (generated for the map data 108 to be analysed). This may for example be done by Kosaraju's algorithm or Tarjan’s algorithm.

[0049] The SCC classes are designed to indicate what (if any) connectivity issues may be present with that SCC. In an ideal routing graph, there would only be one SCC. Therefore, the SCC classes are a hint towards what is preventing this from being true.

[0050] The SCC class of an SCC is for example determined by the size of the SCC and its connectivity to other SCCs of the routing graph. This breaks the SCC down into different classes (essentially a simple decision tree). An example for the SCC classes is as follows:

- largest SCC: main graph
- totally disconnected SCC: separate graphs (usually no routing problem but can be verified). Separate graphs may be caused for example by islands or private property.
- single SCC: SCCs that are made of single nodes, often forming a line of SCC, can be further decomposed into start/end/isolated based on their in and out connections
- other: unclassified, need more verification.

Other properties by be included in assigning the SCC class to an SCC.

[0051] An example of an algorithm in pseudo-code to classify SCCs into SCC classes is given as follows.

```
def n_i = number of nodes in the SCC
def o_i = number of out edges from SCC
def i_i = number of in edges from SCC

- if n_i = max_{over graph}(n_i): largest
- else if n_i == 1:
```

- if o_i > 0 and i_i == 0: single-start
- else if o_i == 0 and i_i > 0: single-end
- else: single-isolated

- else if o_i == 0 and i_i == 0: multi-isolated
- else: multi-other

[0052] In addition to the overall classifications of the routing graph's SCCs, connectivity relationships are determined. These relationships relate SCCs to other SCCs and provide a rating in a number of categories to indicate how likely these two SCCs have missing connectivity between them.

[0053] For calculating the connectivity relationships, the processor 110, for example, collects statistics on each individual SCC, and determines how these statistics relate to the wider routing graph and obtains driver travel history data. It may also use these statistics to build the connectivity classification for each SCC.

[0054] The connectivity relationships are computed between SCCs (wherein the connectivity between SCCs is weaker than strongly connected). For example, the connectivity of two or more SCCs is given a connectivity (or "secondary") relationship as follows

- connectivity relationship "connected": scc_i and scc_j are connected if there are any edges between them
- connectivity relationship "path": list of {scc_i} are a path if they are all single scc and scc_i is connected to scc_{i+1} for each scc in the path, and no other single scc can be added
- connectivity relationship "driver-connected-paths": two paths {scc_i} {scc_j} are driver-connected-paths if we have historic data of a driver travelling between nodes in the two paths, but they are otherwise disconnected.

[0055] This means that after determining the connectivity relationships between pairs or groups of SCCs, SCCs are related to one another by taking advantage of their proximity, shared relationships, graph cuts and historic data (e.g. from historic driver trajectories) to rate the probability of there being a missing connectivity between them. The trajectory data allows (in some cases) to identify suggested locations for the connectivity.

[0056] According to various embodiments, the isolation categorization and detection is built into a pipeline of operations to process, classify, generate improvements as illustrated in FIG. 4. In the following, this is described as being performed by the server computer 106 of FIG. 1

but this is only an example and it may also be performed elsewhere, e.g. by another server computer (e.g. in a map data processing centre etc.) which provides the processed map data then to the server computer 106 for serving e-hailing requests.

[0057] FIG. 4 shows a flow diagram 400 according to an embodiment.

[0058] In 401, current map data is downloaded from a map repository, e.g. external database 111. This could either be a public or private repository, and both may be used. The processor 110 then generates a routing graph from the map data (e.g. for a certain geographic region) wherein it maps geographical locations to vertices and includes a directed edge into the routing graph form one vertex to another, if, according to the map data, there is a road from the vertex to the other vertex.

[0059] The processor 110 then processes the routing graph according to a classification process. This includes the determination of SCCs of the routing graph in 402, the determination of SCC classes for the determined SCCs in 403 and the determination of connectivity relationships for graphs in 404 as described above. This may in particular include, once all of the SCC components are found, collection of information about each SCC and its usage to categorize it into different types of potential map issues. The results of this may be collected and exported.

[0060] In 405, following the classification, the quality of the routing graph may be assessed based on the number of issues found and their severity. This may include determining a quality indicator, e.g. in the form of the value of a metric like

scc_count/node_count

i.e. the ratio of strongly connected components to the number of vertices (or nodes) in the routing graph. This example of a metric is simple but meaningful and gives a problem size estimate independent of map size – it should be a low number in the case of a high quality routing map.

[0061] The quality indicator may be exported (e.g. to an observation tool) and compiled into a time series. This allows map quality to be tracked over time, and gives an early warning for any map quality downgrades.

[0062] The output of the classification process can be exported, for example in form of one or more GPX (GPS Exchange Format) files. GPX is described by an XML schema and is designed as a common GPS (Global Positioning System) format for software applications. This can for example be fed into a visualization tool (e.g. OSM editor) for visualization (e.g. to be

inspected by a user). With the aid of the connectivity classification, this allows (e.g. a user or an automatic analyzation tool) to identify many map errors caused by missing roads, wrong road and turn restrictions, and roads having incorrect direction of travel. Following the detection of an error, one or more changes to the map data may be submitted (e.g. by a user or automatically) and the process may again start with 401 with the changed map data. Each fix of the map data in this manner improves the classification of problems and thus forms a feedback improvement loop. This series of operations provides a closed loop system, resulting in iteratively improving maps.

[0063] The visualisation of the classification process results may include displaying the different classes and connectivity relationships by colour coding. For example, a user (e.g. of a map operations team of an e-hailing service) may overlay a visualization of the detections with the routing graph and satellite imagery. The above connectivity relationships can be tagged with suggested issues such as

- path: carpark, parking lot
- driver-connected-path: carpark, missing two-way street
- multi-isolated: missing bridge, incorrect travel direction

These may be based on experience. An automated feedback into categories may also be provided.

[0064] FIG. 5 shows an example visualization of the result of the processing of map data.

[0065] In this example, the roads belonging to largest SCC is shown with white filling and roads which are part of other SCCs are shown with black filling.

[0066] It should be noted that according to one embodiment map data is separated into multiple regional maps and a distributed system is implemented to handle the regional maps in parallel. As map data (e.g. city level maps) are added, jobs to run the processing on those graphs are may be added to a processing queue by a coordinator. Then, jobs are pulled from this queue and processed, e.g. as described with reference to FIG. 4.

[0067] FIG. 6 shows an example where a poor or outdated satellite image leads to map data which gives rise to SCCs which are disconnected from the largest SCC in the routing graph. It is possible to identify where the map data (supposedly) lacks roads (which are there in reality) resulting in the disconnected SCCs.

[0068] FIG. 7 shows an example wherein the map data misses a travel possibility between two lanes in two cases: the routing graph lacks a connection between the lane going to the

lower left and the lane going to the upper right in the locations 701 and 702. From these two missing connections it can be deduced that the map data lacks, in each case, the travel possibility between the two lanes (e.g. a piece of road connecting the two lanes or the right to turn at the locations 701 and 702).

[0069] In summary, according to various embodiments, a method is provided as illustrated in FIG. 8.

[0070] FIG. 8 shows a flow diagram 800 illustrating a method for correcting errors in map data.

[0071] In 801, map data specifying travel possibilities between locations is obtained.

[0072] In 802, a routing graph is generated from the map data by assigning a vertex to each location and an edge from one vertex to another vertex if the map data specifies that it is possible to travel from the location to which the first vertex is assigned to the location to which the second vertex is assigned.

[0073] In 803, the largest strongly connected component of the routing graph is identified.

[0074] In 804, one or more further strongly connected components of the routing graph are identified.

[0075] In 805, errors in the map data are detected by identifying travel possibilities that are not in the map data which lead to missing connections between the one or more further strongly connected components of the routing graph and the largest strongly connected component of the routing graph.

[0076] In 806, the map data is amended by the identified travel possibilities.

[0077] According to various embodiments, in other words, errors in a (geographic) map are determined using the strongly connected components of the corresponding routing graph as a basis. This can be seen to be based on the observation that even though routing failures occur from time to time, most places (locations) are usually routable from other places. That means that these places form an SCC. By identifying this SCC and placing it on the map it can be seen why the remaining parts of the map (i.e. locations) are not part of the SCC.

[0078] According to various embodiments, missing connections between SCCs are investigated, in particular between the further SCCs and the largest SCC, to find errors (i.e. lack of travel possibilities) in the map data. In other words, the lack of connectivity between the SCCs and the largest SCC is investigated to find errors in the map data.

[0079] The method of FIG. 8 may be part of a method for controlling a vehicle (or controlling the navigation of a vehicle) wherein the amended map data is provided to a vehicle and the vehicle (or its navigation) is controlled using the amended map data.

[0080] The method of FIG. 8 is for example carried out by a server computer as illustrated in FIG. 9.

[0081] FIG. 9 shows a server computer 900 according to an embodiment.

[0082] The server computer 900 includes a communication interface 901 (e.g. configured to receive map data to be processed (i.e. amended or enhanced) and/or to provide processed map data to another server (e.g. a navigation server) or a navigation device in a vehicle). The server computer 900 further includes a processing unit 902 and a memory 903. The memory 903 may be used by the processing unit 902 to store, for example, map data to be processed. The server computer is configured to perform the method of FIG. 8.

[0083] The methods described herein may be performed and the various processing or computation units and devices described herein may be implemented by one or more circuits. In an embodiment, a "circuit" may be understood as any kind of a logic implementing entity, which may be hardware, software, firmware, or any combination thereof. Thus, in an embodiment, a "circuit" may be a hard-wired logic circuit or a programmable logic circuit such as a programmable processor, e.g. a microprocessor. A "circuit" may also be software being implemented or executed by a processor, e.g. any kind of computer program, e.g. a computer program using a virtual machine code. Any other kind of implementation of the respective functions which are described herein may also be understood as a "circuit" in accordance with an alternative embodiment.

[0084] While the disclosure has been particularly shown and described with reference to specific embodiments, it should be understood by those skilled in the art that various changes in form and detail may be made therein without departing from the spirit and scope of the invention as defined by the appended claims. The scope of the invention is thus indicated by the appended claims and all changes which come within the meaning and range of equivalency of the claims are therefore intended to be embraced.

CLAIMS

1. A method for correcting errors in map data comprising:

obtaining map data specifying travel possibilities between locations;

generating a routing graph from the map data by assigning a vertex to each location and an edge from one vertex to another vertex if the map data specifies that it is possible to travel from the location to which the first vertex is assigned to the location to which the second vertex is assigned;

identifying the largest strongly connected component of the routing graph;

identifying one or more further strongly connected components of the routing graph;

detecting errors in the map data by identifying travel possibilities that are not in the map data which lead to missing connections between the one or more further strongly connected components of the routing graph and the largest strongly connected component of the routing graph;

amending the map data by the identified travel possibilities; and

iteratively amending the map data by iteratively repeating the generating of a routing graph, identifying the largest strongly connected component of the routing graph, identifying one or more further strongly connected components of the routing graph, detecting errors and amending the map data, wherein the map data amended in a previous iteration is used as map data for a current iteration.

2. The method of claim 1, wherein the travel possibilities are directed travel possibilities and the routing graph is a directed routing graph.

3. The method of claim 1 or 2, wherein the travel possibilities comprise roads associated with an allowed travel direction.

4. The method of any one of claims 1 to 3, comprising associating each further strongly connected component of the routing graph with a connectivity class of a plurality of connectivity classes, wherein the connectivity classes comprise one or more of a class

of strongly connected components each being isolated from the largest strongly connected component and comprising multiple vertices, a class of strongly connected components each having a single vertex isolated from the other strongly connected components, a class of strongly connected components each having a single vertex which can only be reached from a vertex of one or more other strongly connected components, and a class of strongly connected components each having a single vertex which cannot be reached from any other strongly connected component but from which strongly connected component can be reached, wherein the method further comprises detecting errors in the map data using the connectivity classes associated with the further strongly connected components of the routing graph.

5. The method of any one of claims 1 to 4, comprising associating groups of two or more strongly connected components of the routing graph with a connectivity relation of a plurality of connectivity relationships, wherein the connectivity relationships comprise one or more of a connectivity relationship indicating that there are edges between the strongly connected components of the group, a connectivity relationship indicating that the group of strongly connected components forms a maximum path of single vertices and a connectivity relationship indicating that historic driving data of a geographical region represented by the map data indicates that a driver has travelled between vertices of the strongly connected components which are not connected in the routing graph, wherein the method further comprises detecting errors in the map data using the connectivity relationships associated with the groups of strongly connected components of the routing graph.

6. The method of any one of claims 1 to 5, further comprising acquiring historic driving data of a geographical region represented by the map data, determining whether there exists a pair of vertices comprising a first vertex and a second vertex in the routing graph wherein the map data does not specify a travel possibility form the first vertex to the second vertex but the historic driving data indicates that a driver has travelled from the location to which the first vertex is assigned to the location to which the second vertex is assigned and determining that the map data lacks a travel possibility

from the location to which the first vertex is assigned to the location to which the second vertex is assigned if such a pair of vertices exists.

7. The method of any one of claims 1 to 6, comprising generating an alarm signal when the quality indicator falls below a predetermined threshold.

8. The method of any one of claims 1 to 7, comprising exporting a specification of the identified strongly connected components of the routing graph to a visualization tool.

9. The method of any one of claims 1 to 8, further comprising visualizing the identified strongly connected components of the routing graph.

10. The method of claim 9, wherein the visualizing further comprises visualizing connectivity classes of strongly connected components or connectivity relationships of groups of two or more strongly connected components or both.

11. The method of claim 9 or 10, wherein the visualizing further comprises overlaying the strongly connected components with a geographic map given by the map data.

12. The method of any one of claims 1 to 11, further comprising navigating one or more vehicles by the amended map data.

13. The method of any one of claims 1 to 12, further comprising transmitting the amended map data to a vehicle and controlling the vehicle using the amended map data.

14. A server computer comprising a communication interface, a memory and a processing unit configured to perform the method of any one of claims 1 to 13.

15. A computer program element comprising program instructions, which, when executed by one or more processors, cause the one or more processors to perform the method of any one of claims 1 to 13.

16. A computer-readable medium comprising program instructions, which, when executed by one or more processors, cause the one or more processors to perform the method of any one of claims 1 to 13.