

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; and
amending the map data by the identified travel possibilities.
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 from 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 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.

8. The method of any one of claims 1 to 7, comprising 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.
9. The method of claim 8, comprising generating an alarm signal when the quality indicator falls below a predetermined threshold.
10. The method of any one of claims 1 to 9, comprising exporting a specification of the identified strongly connected components of the routing graph to a visualization tool.
11. The method of any one of claims 1 to 10, further comprising visualizing the identified strongly connected components of the routing graph.
12. The method of claim 11, 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.
13. The method of claim 11 or 12, wherein the visualizing further comprises overlaying the strongly connected components with a geographic map given by the map data.
14. The method of any one of claims 1 to 13, further comprising navigating one or more vehicles by the amended map data.
15. The method of any one of claims 1 to 14, further comprising transmitting the amended map data to a vehicle and controlling the vehicle using the amended map data.

16. A server computer comprising a radio interface, a memory interface and a processing unit configured to perform the method of any one of claims 1 to 15.
17. 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 15.
18. 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 15.