AUTOMATIC CHECKING OF ROAD NETWORK MODELS

I. Abdulganiev, A. Agafonov

Samara National Research University, Samara, Russia

Abstract. This paper addresses the task of automatic checking of the road network model. Under the road network model checking, we mean checking completeness and topology of the road network. We propose an original algorithm based on a direction of movement criterion. The proposed algorithm is compared with an existing algorithm based on an optimal distance criterion. The algorithm was implemented and tested using the large transport network of Samara, Russia.

Keywords: transport network, topology correctness, shortest path, movement direction.

1   Introduction

Geoinformation systems are widely used in many areas of our life, including municipal management, transport systems, economics, etc. The question about quality of spatial data becomes more and more important because of increase in the intensity of their usage. In this paper under quality we mean completeness and correctness of the data.

Road network model is the basis for many transport problems, such as the optimal network planning, optimization of the public transport routes, navigation problems and so on. The paper deals with one of the possible formulations of the road network checking tasks: checking connections between road links and correctness of turns between links. The solution to this problem is necessary for the further use of the road network in more complex problems, such as shortest path calculation and reliable routing in stochastic networks [1], public transport arrival time prediction [2], traffic flow forecasting [3], etc.

Completion of incompletely extracted road networks is an important part of road network model with the use of digital imagery [4, 5, 6], including hyperspectral images [7]. Road networks automatically extracted from digital imagery are often incomplete and fragmented [8, 9]. This problem is especially important for the extraction from low-resolution images. Mainly the operator, who compares the vector data and the

source images, carries out the network checking [10]. Completeness and topology of the extracted network can be improved by the use of the global network structure [11].

The article [4] proposes an approach to complement the fragmented road networks by examining link hypotheses between nodes on the network, which are likely to be connected, based on the characteristics of the network. In paper a detour factor, which taking into account the distance of the road network and the optimal (Euclidean) distance between points, is checked. The shortest path algorithm A* is used to search for the missing road links of the network in [5].

In this paper we use a complex road network graph, that includes the permitted / prohibited turns between road links. We propose a road network checking algorithm based on a direction of movement factor. The proposed algorithm is compared with an existing algorithm based on detour factor and described in [4].

The paper is structured as follows. In the second section we introduce the basic notation, describe the graph structure and give a problem statement. Existing and proposed algorithms are described in sections 3 and 4. In the fifth section we present the experimental results. Finally, we make conclusions and provide recommendations for further work.

2   Basic notations and problem statement

We consider a road network as an oriented graph $G = (V, E)$, where V is a set of nodes corresponding to the road intersections, E is a set of edges corresponding to the road links. Direction of the edge is defined by the driving direction at the corresponding road link.

We assume that the graph has a spatial reference, i.e. each node of the graph has the coordinates (x, y) that are determined by the location of the corresponding road intersection in the real road network.

In this paper we consider an oriented graph, each edge of which has two weight coefficients:

— t is the travel time on the edge;
— l is the length of the edge.

Each intersection is converted into a set of geometrically coinciding graph nodes. Each road link is presented by a pair of graph edges with different directions. We add new edge between pair of nodes, if a turn between this nodes is allowed. Additional edges do not have geometric characteristics, their weights are defined by average travel time. Figure 1 shows a diagram of the intersection with the possibility of movement in any direction.

The formal statement of the road network model checking problem can be defined as follows: develop and implement algorithms for checking the topology of a large city road network.

We check the connectivity of nodes of the road network and correctness of turns between road links. The following sections provide algorithms for checking link hypotheses between nodes in the network, based on a detour factor and a movement direction factor.
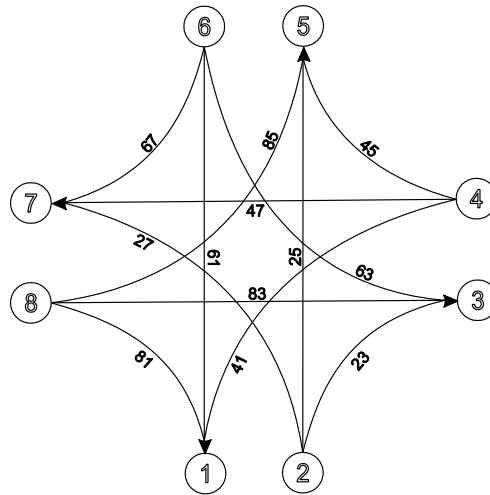

Fig. 1. Graph structure inside the junction

3   Hypothesis based on a detour factor

In the paper [1] preliminary link hypotheses are defined between each pair of nodes. A detour factor is calculated for each preliminary link hypothesis.

Let σ_{ks} be the shortest path distance between nodes k and s. Shortest paths are calculated using Dijkstra algorithm [12];

ρ_{ks} be the optimal (Euclidian) distance between nodes k and s;

$\theta_{dc} = \dfrac{\sigma_{ks}}{\rho_{ks}}$ be a detour factor.

The detour factor shows the ratio of the distance between nodes in a road network to the optimal distance between these nodes.

For each pair of nodes involved in checking we calculate distances σ_{ks} and ρ_{ks}. Then we check the following criterion:

$$\theta_{dc} > \theta_{dc}^{\max} \tag{1}$$

where $\theta_{dc}^{\max}$ is predetermined threshold coefficient.

If the criterion (1) is satisfied, then all the links of the road network, included in the shortest path between two nodes, is the candidates for additional check, further study is needed for their analysis.

Experiments, how the road network checking results depend on the threshold coefficient $\theta_{dc}^{\max}$, are presented in section 4.

4   Hypothesis based on a direction of movement factor

Algorithm based on the detour factor has a drawback. If the Euclidian distance between pair of nodes is large, then the distances σ_{ks} and ρ_{ks} will be almost equal, even if the connectivity of road links is broken. In this section we propose an algorithm based on a different hypothesis, which eliminates this drawback.

Let $d_{ks} = \{v_1, v_2, \ldots, v_n\}$ be the shortest path from the node k to the node s, where

$v_i, i = \overline{0, N-1}$ is the node in the shortest path with index i.

N is a number of nodes in the shortest path.

The proposed algorithm consists of the following steps:

1. For pair of nodes k and s shortest path d_{ks} is calculated.

2. For each node $v_i \in d_{ks}, i = \overline{0, N-1}$ the optimal (Euclidian) distance $\rho_{v_i s}$ is calculated.

3. Direction of movement factors $\beta_i, i = \overline{0, N-2}$ are calculated using the following expression:

$$\beta_i = \frac{\rho_{v_{i+1}s}}{\rho_{v_i s}}, v_i \in d_{ks}, i = \overline{0, N-2}.$$

4. The following criterion is checked:

$$\exists i : \beta_i > 1, i = \overline{0, N-2}. \tag{2}$$

If the criterion (2) is satisfied, then all the links of the road network, included in the shortest path between two nodes, is the candidates to check. The criterion (2) means that the Euclidian distance to the destination node should decrease during the trip.

Such criterion also has a drawback: it can gives false positive error if the optimal distance is slightly increased. To avoid this, we introduce threshold coefficient $\beta^{\max}$. So the criterion is modified as follows:

$$\exists i : \beta_i > \beta^{\max}, i = \overline{0, N-2}. \tag{3}$$

The next section presents the conducted experiments.

5   Test results and analysis

We provide the following experiments.

- Experiment 1. Study of the threshold coefficient $\theta_{dc}^{\max}$ impact on the results of the algorithm based on the detour factor.
- Experiment 2. Experimental study of the algorithm based on the direction of movement factor.

The algorithms are tested on the road network of Samara, Russia. The network consists of 52175 nodes and 76636 links. For experiments 400 graph edges $\{e_k\} \in E$ has been artificially removed. The experiment results are presented below.

5.1   The impact of the threshold on the detour based algorithm results

For this experiment we choose 10 origin nodes $(k_1, k_2, ..., k_{10})$ and 10 destination nodes $(s_1, s_2, ..., s_{10})$, located in different areas of the city. We choose the following threshold coefficient values $\theta_{dc}^{\max} = \{1.08, 1.15, 1.2, 1.3, 1.35, 1.4\}$.

Firstly for each pair of nodes k, s we calculate the shortest path d_{ks} in the original graph (without removed edges). Next step is performed using the changed graph with removed edges $\{e_k\}$. For each pair of origin-destination nodes we calculate the shortest path distance σ_{ks}, the optimal distance ρ_{ks} and the detour factor θ_{ks}.

If the shortest path d_{ks} in the original graph contains some edges $e \in \{e_k\}$, and the criterion (1) for the changed graph is satisfied, then we suppose, that removed edges is found. If the criterion (1) is satisfied, but the shortest paths in the original and changed graphs are equal (i.e. the shortest path does not contain removed edges), then road network check completed with error. The number of successfully found edges and the errors number are shown on figure 2.
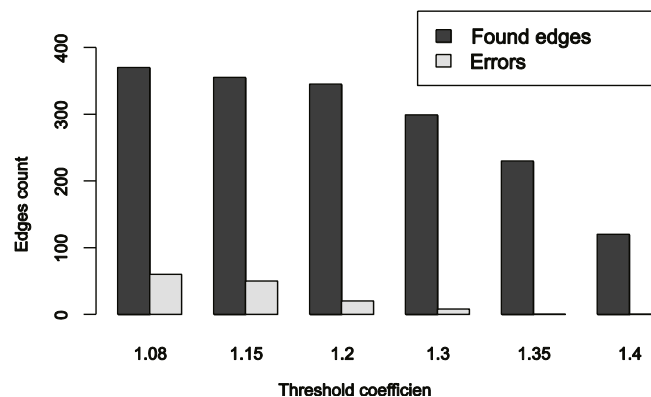

Fig. 2. The impact of the threshold on the detour based algorithm results

Chosen value of the threshold coefficient $\theta_{dc}^{\max}$ depends on the allowable number false positive errors of the algorithm. Using the threshold value in the range $[1.2, 1.35]$, we detect about 75-85% of the road network topology errors. Level of incorrectly defined edges is about 4-6%.

5.2   Experimental study of the algorithm based on the direction of movement factor

In these experiments we use the same graph and pairs of origin-destination nodes, defined in the previous section.

Again, for each pair of nodes k, s we calculate the shortest path d_{ks} in the original graph, the shortest path d_{ks}' in the changed graph and the direction of movement factors $\beta_i, i = \overline{0, N-2}$.

If the criterion (2) for the pair of origin-destination nodes k, s is satisfied and the shortest path d_{ks} contains removed edges $e \in \{e_k\}$, then the algorithm completed successfully. If the criterion (2) is satisfied, but the shortest path d_{ks} does not contain removed edges $e \in \{e_k\}$, then the algorithm completed with false positive error. The results of the experiment are shown in table 1.

Table 1. The results of the algorithm based on the direction of movement factor

Removed edges count	Found edges	False-positive errors
400	354	22

The proposed algorithm based on the direction of movement factor shows the similar results with the algorithm based on the detour factor with the threshold coefficient value $\theta_{dc}^{\max} = 1.2$. We detect 88% of the removed edges with false-positive errors in 5.5%.

6   Conclusions

In this paper we propose the algorithm for road network checking based on the direction of movement factor. Under road network checking we mean checking the connectivity of the road links and correctness of turns between links. The proposed algorithms was compared with the base algorithm based on the detour factor. The conducted experiments was performed using the road network of Samara, Russia. The proposed algorithm showed similar results with the base algorithm with the threshold coefficient value $\theta_{dc}^{\max} = 1.2$. It should be noted, that the proposed algorithm showed good results even in the case, when Euclidian distance between origin and destination nodes are large enough.

We plan to conduct the experiments, how the threshold coefficient value β^{max} influence on the results of the algorithm based on the direction of movement factor.

Acknowledgements

References

1. Agafonov A, Myasnikov V. Method for the reliable shortest path search in time-dependent stochastic networks and its application to GIS-based traffic control. Computer Optics, 2016; 40(2): 275-283 [in Russian]. DOI: 10.18287/2412-6179-2016-40-2-275-283.
2. Agafonov A, Myasnikov V. An algorithm for city transport arrival time estimation using adaptive elementary predictions composition. Computer Optics, 2014; 38 (2): 356-368. [in Russian]
3. Agafonov A, Myasnikov V. An algorithm for traffic flow parameters estimation and prediction using composition of machine learning methods and time series models. Computer Optics, 2014; 38 (3): 539-549. [in Russian]
4. Wiedemann C, Ebner H. Automatic Completion and Evaluation of Road Networks. International Archives of Photogrammetry and Remote Sensing, 2000; XXXIII, Part B3: 979-986.
5. Gerke M, Butenuth M, Heipke C, Willrich W. Graph Supported Automated Verification of Road Databases Using Aerial Imagery. Proc. 2nd International Symposium on Spatial Data Quality, 2003: 421-430.
6. Grote A, Heipke C, Rottensteiner F. Road Network Extraction in Suburban Areas. Photogrammetric Record, 2012, 27 (137): 8-28.
7. Kuznetsov A, Myasnikov V. A comparison of algorithms for supervised classification using hyperspectral data. Computer Optics, 2014; 38 (3): 494-502.
8. Wiedemann C, Heipke C, Mayer H, Jamet O. Empirical Evaluation of Automatically Extracted Road Axes. Empirical Evaluation Methods in Computer Vision, 1998: 172-187.
9. Steger C, Mayer H, Radig B. The role of grouping for road extraction. Automatic Extraction of Man-Made Objects from Aerial and Space Images (II), Verlag, Basel, Switzerland, 1997: 245–256.
10. Willrich F. Quality Control and Updating of Road Data by GIS-driven Road Extraction from Imagery. International Archives of Photogrammetry and Remote Sensing, 2002; 34: 761-767.
11. Mayer H, Laptev I, Baumgartner A, Steger C. Automatic road extraction based on multiscale modeling, context, and snakes. In: International Archives of Photogrammetry and Remote Sensing, 1997; 2: 106–113.
12. Dijkstra EW. A note on two problems in connexion with graphs. Numer. Math – Springer Science+Business Media, 1959; 1(1): 269-271.