Intelligent Vehicular Navigation System Based on Real-time Road Information

Hsien-Tsung Chang[1] and Sheng-Yuan Chen[2]

[1]Chang Gung University, smallpig@widelab.org
[2]Chang Gung University, styuan@widelab.org

ABSTRACT

Taiwan is a densely populated island, thus, main roads and roads along tourist attractions are often highly congested during rush hours, weekends, and national holidays. The traffic conditions would waste road users' time and affect the mood of relaxation during holidays. Navigation system is an indispensable tool for many road users; however, the traditional navigation system with the RDS-TMC system can't provide real-time road information for dynamic route planning because of the data capability of RDS-TMC is not large enough carry all information of every road. When non-primary roads are congested, the systems cannot reroute for the road users. As most users use the same route planning, serious congestion would also occur. This paper intended to propose a method to reroute road users according to actual road conditions, thus assisting road users to avoid congestion, and balance road use. The method can greatly reduce unnecessary time waste and try to make the road usage in an average manner. This paper includes two parts: (1) data collection and integration; (2) dynamic route planning based on the collected data. Using the concept of Web 2.0 to collection and integration data, real-time data was provided by road users. The real-time road information is establishing with Internet and road users, and completed by some estimation based on the history information.

Keywords: vehicular navigation system, real-time road information, navigation system.
DOI: 10.3722/cadaps.2012.207-214

1 INTRODUCTION

Taiwan is a densely populated island, thus, main roads and roads along tourist attractions are often highly congested during rush hours, weekends, and national holidays. The traffic conditions would waste road users' time and affect the mood of relaxation during holidays. Navigation system is an indispensable tool for many road users. With the development of science and technology, the cost of the navigation system is reduced, and the system has become more prevalent in daily use. Compared to the embedded navigation system, portable navigation system can better meet users' demands. Therefore, more and more manufacturers are engaged in the research and development of the navigation system.

To enhance the competitiveness of the systems, manufacturers have introduced new targets and functions. For example, a navigation system can provide route planning based the users' driving time, driving distance, driving comfort, congested roads, location of tolls, and alternative route. Therefore, the navigation system must acquire more precise data, provide real-time communication capability, precision in computing for the optimal route, and have a large database to record history data.

All navigation systems available in the market need to acquire information before route planning. Although, the method is theoretically acceptable, sudden events may cause changes in road conditions, and the planned route is no longer optimal. For example, when an accident occurs in one section of the optimal route planned before departure, the passing efficiency of the section will be greatly reduced. Many navigation systems, which cannot provide timely modification, may still regard the section as a high-efficiency section, and guide the users to the section, thus further reducing passing efficiency of the section. Thus, a navigation system that does not have real-time information may make road conditions worse.

As the navigation system has more functions, powerful hardware is needed; however, the mobile navigation system has its limitations. The general solution is to choose between computing time and accuracy of optimal route, i.e. to compute an approximate optimal result within the shortest time. This study intended to design a set of algorithms to compute the optimal solution for route planning within the minimum time. The route can be modified according to the changes in road conditions, thus avoiding worsening the road conditions.

2 RELATED WORKS

In order to address the above problems, previous researchers have proposed suggestions on data collection and algorithm improvement.

In terms of data collection, the navigation system is divided into two types, namely centralized and autonomous. The former has a powerful information center, which is responsible for information collection, integration and distribution. With this information center, the mobile device requires less memory and computing capability. The latter device has its own database, and only collects useful information and stores the information in the mobile device. However, both of these methods cannot cover all real-time information, so the Prediction module is added to infer possible road conditions in the areas with insufficient information.

In terms of algorithm improvement, distinct modifications are made according to different demands based on basic Dijkstra algorithm. The algorithms A* algorithm and bidirectional navigation reduce the computing time, map pruning and hierarchical level map that pre-process collected data to reduce amount of computation, and virus genetic algorithm and adaptive navigation that can modify dynamic route planning of original route according to the changes in road conditions.

A* algorithm is based on Dijkstra. With the estimated value of the distance between representative node and end node, each node has two values, which are the real distance from start node to present node and the estimated value from present node to end node. In route planning, the node with a smaller sum of the two distances is chosen as the next node. Bidirectional navigation [2] is based on both Dijkstra and A* algorithm, while its difference from the two algorithms is that the two algorithm begin from the start point or end point for route planning and end at the other one, but Bidirectional navigation begin from both. If there is only one optimal route, then route planning from two directions may meet between the start point and end point, and the required computing time is only half of original computing time. However, if two routes, which do not intersect at all, are computed, the consumed time will be the same as original time for one direction.

In pre-processing, after map information is acquired, the part that can be computed for future route planning is computed, so as to reduce the amount of computation in future route planning. Many current studies have presented effective methods. Map pruning pre-processes the map, to reduce the map complexity and the amount of computation in the execution of route planning. Map pruning may involve short cut and reach base pruning. Short cut [2] reduces two connected roads into one visual road. If one route runs from A to C via B As shown in Fig. 1, it will be reduced into a visual route from A to B, and its length is $Dist_{AB}$ + $Dist_{BC}$. For Reach base pruning [2], as shown in Fig. 2, if several routes pass through V, the longest one from shorter sub-path is chosen as R (V). Prior to route planning, R value of all points is calculated. In order to determine the route between two points, if a sub-path through V is smaller than R (V), V may be relay point of these two points; vice versa. To reduce the amount of computation, hierarchical level map divides all nodes into two kinds, namely high level and low level. High level nodes are all important ones, such as important traffic hubs, while others are set as low level nodes. Before detail route planning, all high level nodes are calculated, and then important traffic hubs to be passed are planned, followed by planning of sub-paths between these traffic hubs. As shown in Fig. 3, the line is used to denote obstacle, and both end points of the line are taken as high level nodes [1]. The complexity of former Fig. 3 (a) is reduced into that of Fig. 3 (b) after computation of high level nodes.

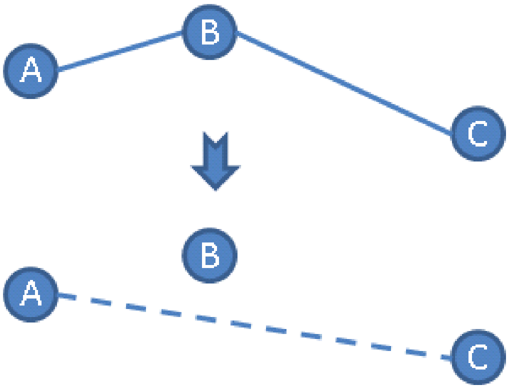


Fig. 1: Create a virtual short cut replace the original route. [2]

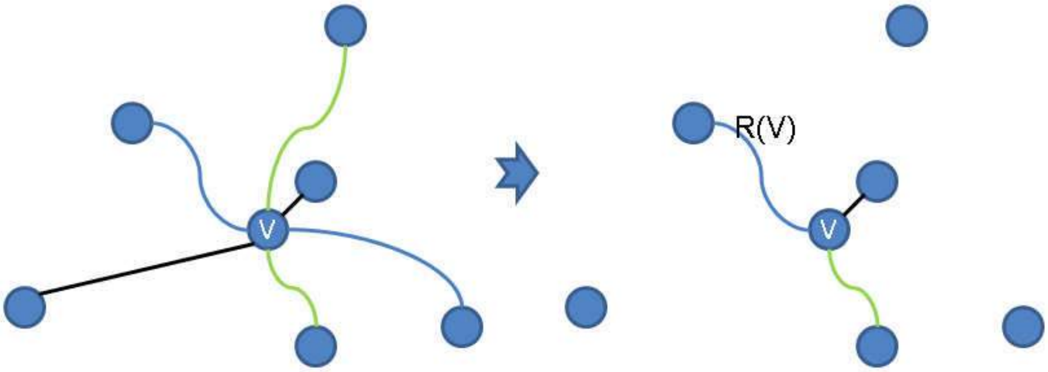


Fig. 2: Each color represents a route. Select the shortest sub-path from each routes, which through node V, and choice the longest one as R(V). [2]

In virus genetic algorithm [4], after an optimal route is planned, as shown in Fig. 4 (a), if a new route is found intersecting with the route at two points, as shown in Fig. 4 (b), and the quality of the original route may be raised if these two intersection points are switched to the new one, then the old route with poorer quality will be rejected, as shown in Fig. 4 (c). Adaptive navigation [5] collects information from the former vehicle, and infers the required time if the route is chosen, based on driving speed and time passing the route of the former vehicle and driving speed of our own vehicle, so as to select an optimal route.

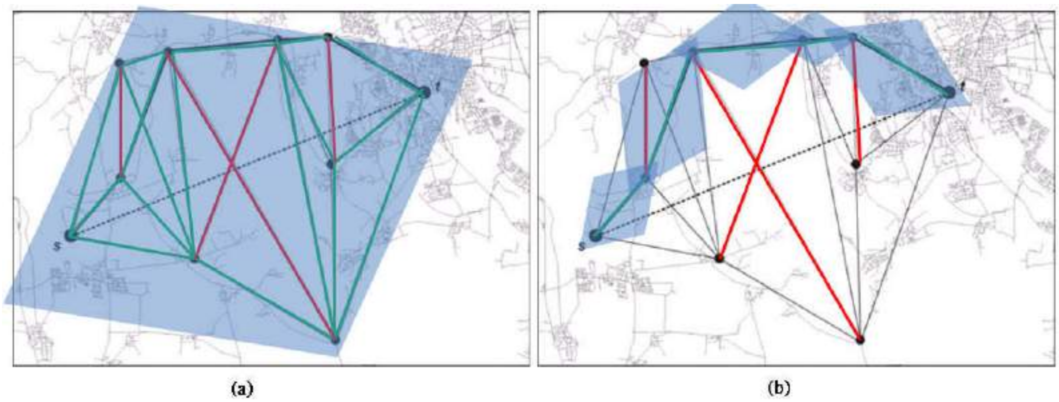


Fig. 3: Hierarchical level map could reduce the complexity of a complicated map. The red lines represent the obstacle in this map, select all the endpoints of the red line as high level nodes. [1]

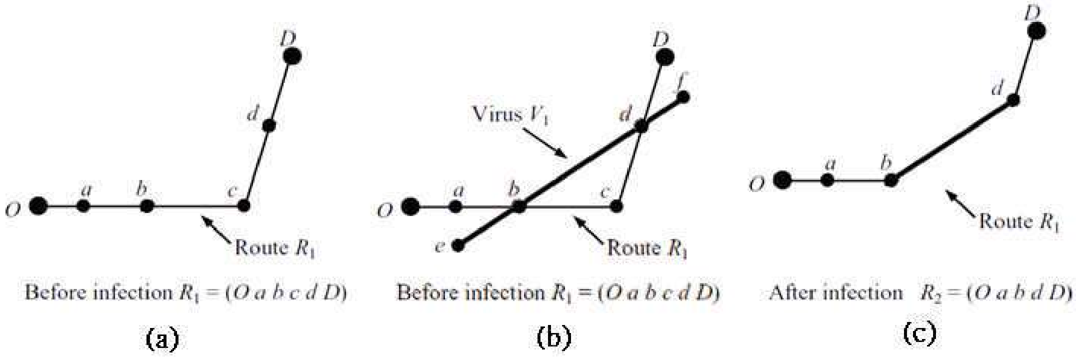


Fig. 4: Infect the original route by the better route to increase its quality. [4]

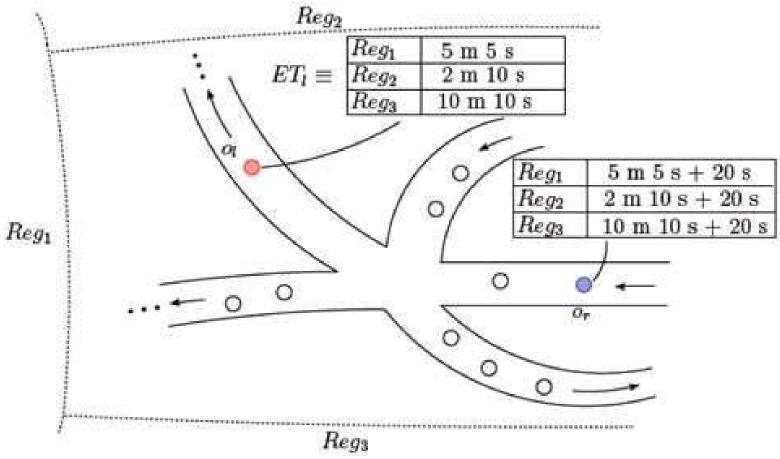


Fig. 5: Pick up the best selection by estimate the time of the chosen route will take. [5]

3 OUR POROPOSED METHOD

In related works, the weights of edges are determined according to distance or time, but this cannot exactly reflect the influence of dynamic information. In this paper, the weights are also determined by time, but the time is estimated by Eqn. 3.1, in which V_{est} is different from other research which using road limit, it is the estimated speed in the road section. The V_{est} is base on real-time car velocity, so it can represent real situation. The details will be explained later. The proposed method includes two parts: (1) data collection and integration; (2) dynamic path planning in line with the extracted information, detailed as follows.

$$W = \frac{\text{Distance}}{V_{est}} \qquad \text{(Eqn.3.1)}$$

3.1 Data Collection and Integration

In this section, Web 2.0 is used to involve users in data collection, thus reducing the workload of the system and ensuring the most updated information. First, a data center is established for (1) collecting the approximate real-time information provided on-line; (2) integrating the information provided by user; (3) estimating the possible data of the road sections lack of information; (4) recording the driving speed at each road section as time progresses.

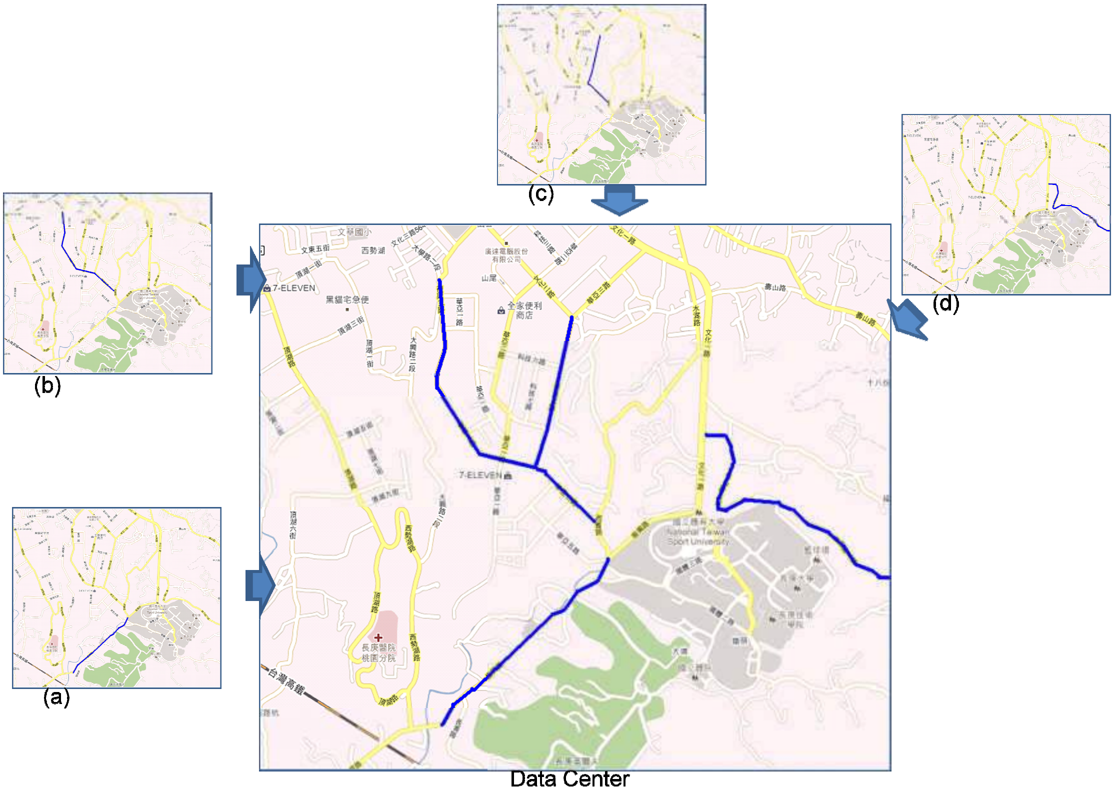


Fig. 6: (a) (b) (c) (d) are representing individual data set from user.

In order to acquire the most updated information, the real-time information on traffic conditions of national highway provided by the Ministry of Transportation of Communications (http://1968.freeway.gov.tw/) is retrieved, and then supplemented by the information of common roads. The mobile device systems of the users are set to record the driving information for every 10 seconds, including the coordinate, driving direction, and driving speed. Every minute, the collected information is transmitted to the data center, as shown in Fig.6, where the information is integrated. In case that a multiple speed data are at the same coordinate and driving towards the same direction, such as Fig. 6 (b) and (c) where a part of collected information overlaps, the mean values are recorded. However, if the collected traffic information is normal, that is, conforming to the speed limit, the archived history date is useless. In order to avoid this situation, the history data only keep the unusual points, such as the place, time and driving speed in the points of congestion. Users in our system are participated the data generating when they consuming the service data from our data center. It is why we call it as a Web 2.0 like mechanism. This mechanism can avoid the kind of users who only want to consume data without providing their data.

Although updated information can be obtained from users' mobile device system and the traffic information center of national highways, the information may not cover the entire island, while no related services served provided online. Thus, the possible driving speed is deduced (as the weight of navigation) based on the road speed limit, as well as the traffic information of front and back sections of the road, and the history data. In Eqn. 3.1, we can calculate the weight of ever road section, and the V_est is represent as the estimated driving speed in the road section, and the calculation is as Eqn. 3.2. V_frontis the driving speed in the front section, V_back is the driving speed in the back road section, D_b is the distance from the estimation point to the back road section, D_f is the distance from the estimation point to the front road section, and D_fbis the distance between the front and back road sections. The method of interpolation is used to calculate the proportion of the two sections in the estimated value. The closer they are to this section, the more important they are. RoadHistory denotes the historical record in the previous time period in the database V_est. If any item is lack of data, RoadLimit of this section is used. Compare the attained value V_estand the RoadLimit, the smaller of which represents the driving speed in this section.

$$V_{est} = \frac{V_{front} \times \frac{D_b}{D_{fb}} + V_{back} \times \frac{D_f}{D_{fb}} + \text{RoadHistory}}{2} \qquad \text{(Eqn.3.2)}$$

3.2 Dynamic Path Planning with Real-time Road Information

In order to fasten our system computing, we choose the hierarchical algorithm for our path planning. For hierarchical path planning, all nodes are created on the map of each junction or intersection of two roads. Firstly, the junctions and intersections of all national and provincial highways are classified as high level nodes, and the other intersections are low level nodes. After one month of use, the frequency of use of all nodes is compared; thus, 10% least-used high level nodes is changed to low level nodes, and are replaced by the same number of the most-used low level nodes.

After selecting the high level nodes, the Voronoi diagraph [10] is plotted as shown in Fig. 7, in which all high level nodes (blue nodes) are taken as the benchmark and the nodes are divided into several parts by the vertical line of all the link lines of each two high level nodes. The speed for each part to adjacent part is denoted by the mean value of the low level nodes (green nodes) in this part. If there is no road at the side, it is expressed as unreachable (speed=0).

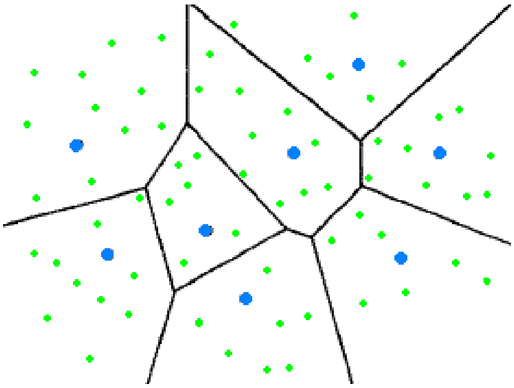


Fig. 7: The blue nodes are high level nodes and the green nodes are low level nodes. Divide the nodes into several parts by the vertical line of all the link lines of each two high level nodes. [10]

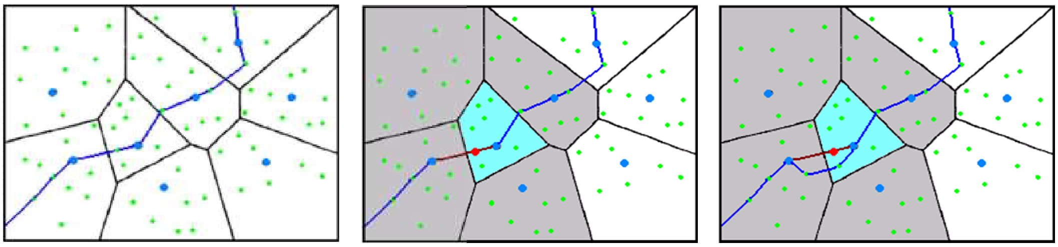
At the time of need for the path planning, the users can acquire the information necessary for the path planning from the data center, plot the optimal path in accordance with the algorithm designed by this paper, and then store the path of each planning into the mobile device.

The proposed algorithm integrates with the hierarchical level map, bidirectional navigation, and virus genetic algorithm. Unlike ordinary bidirectional navigation, different algorithms are used for the two directions, A* starting from the starting node and Dijkstra starting from the destination. Finally, the calculated optimal path is stored in the mobile device as reference for next use.

Congestion Level	*Speed*	*Describe*
1	**S ≤ 10%**	Almost stop
2	10% ≤ S ≤ 30%	Hard to move
3	30% ≤ S ≤ 50%	Slow
4	50% ≤ S	Smooth

Tab. 1: The definition of four levels for the congestion.

When the user what to plan their route, they will transmit a message to the data center, informing the starting point and destination, and acquiring the information about high level nodes between them. After obtaining the information, A* path planning may be done, informing the users of the possible direction of driving and the main traffic junctions to be passed by. The information of the low level nodes between the starting point and the first high level node, and those between the destination and the last high level node, is obtained from the data center. The information is calculated in two processes. Process 1 use A* to calculate the path between the starting node and the first high level node, outputs the calculated results, requests the data center for the information to the next high level node, and repeats these operations until finding out the link point of the computation beginning from the reverse direction, or to the destination. Process 2 is similar to Process 1, but it uses Dijkstra from the destination to seek for the path to the starting point or linked to the result of Process 1. The purpose of using two algorithms for path planning is to have a direction of driving even the computation is ongoing, or start before the computation ends. By using Dijkstra, the characteristics of all paths will be taken into consideration for efficiency in planning.



(a) Route before detected congestion (b) Collection the congestion area and surround area information (c) Re-navigation

Fig. 8: The blue nodes are representing as high level nodes, the green nodes are representing as low level nodes.

Another advantage of utilizing hierarchical navigation is that, when the users find that the path optimization is necessary due to over high Congestion Level in a road section in the previously planned path, only the part of the road section with poor efficacy needs to be recalculated, without re-planning the entire route. The speeds that are slower than the speed limit are classified, as presented in Table 1, where Speed represents the proportion of the driving speed to the speed limit. For instance, suppose that when the user is driving as Fig. 8(a) and the data center receives the information that the Congestion Level of a road section is higher than 3, and transmits it to nearby users, the user determines whether to optimize the path at discretion. If the user wants to optimize the path, the mobile device system will request the data center for the information of the part of this section and adjacent sections as Fig. 8 (b), and then re-navigate as Fig. 8(c). If there are too many parts in need of path optimization, from the time when the traffic information is requested from the data center, where the user locates is regarded as the starting point for re-navigation. The user can also select the Congestion Level of path optimization at discretion.

However, if all road users use the same algorithm to plan their path to a popular scenic spot, this will put all the vehicular in the same road and produce the congestion situation. In fact, there are some other paths with only a little bit larger in the path weight. If we can navigate some of the vehicular to the alternate paths, it will make the roads usage in an average manner. In order to achieve this goal, we add a random vector of a small value to the Eqn 3.1 to become Eqn 3.3. With the ε_{RAND} little random value will make the path with similar total path weight chosen by our navigation system. Of course, if there is only existed on path to the destination, our design is useless.

$$W = \frac{\text{Distance}}{V_{est}} + \varepsilon_{RAND} \qquad \text{(Eqn.3.3)}$$

4 CONCLUSIONS

In this paper, we proposed a mechanism similar to Web 2.0 manner to collect real-time road information. Road users can consume the collected by participating the information generating within their vehicular traveling. By the collected real-time information, we also proposed a hierarchical bi-direction path planning algorithm, and we introduce a new concept of using two different algorithms in the bi-direction computation. We also introduce a roads usage in an average manner by adding a random vector into our algorithm. Future studies can consider other situations, for example, to predict the possible influence area of congestion and the use of this information in dynamic route planning.

REFERENCES

[1] Hahne, F.; Nowak, C.; Ambrosi, K.: Acceleration of the A*-Algorithm for the Shortest Path Problem in Digital Road Maps, Operations Research Proceedings, 2008, 2007(XIX), 455-460. DOI:10.1007/978-3-540-77903-2_70

[2] Goldberg, A. V.: Point-to-Point Shortest Path Algorithms with Preprocessing, Lecture Notes in Computer Science, 2007, 4362/2007, 88-102. DOI: 10.1007/978-3-540-69507-3_6

[3] Cheng, H.; Cao, J.; Fan, X.: GMZRP: geography-aided multicast zone routing protocol in mobile ad hoc networks, Mobile Networks and Applications, 14(2), 165-177. DOI: 10.1007/s11036-008-0135-4

[4] H. Kanoh, Dynamic route planning for car navigation systems using virus genetic algorithms, International Journal of Knowledge-Based and Intelligent Engineering Systems, 11(1/2007), 65-78.

[5] Verroios, V.; Kollias, K.; Chrysanthis, P. K.; Delis, A.: Adaptive Navigation of Vehicles in Congested Road Networks, ICPS '08 Proceedings of the 5th international conference on Pervasive services. DOI:10.1145/1387269.1387277

[6] Lee, J.; Forlizzi, J.; Hudson, S. E.: Iterative design of MOVE: A situationally appropriate vehicle navigation system, International Journal of Human-Computer Studies, 66(3), March 2008, 198-215. DOI:10.1016/j.ijhos.2007.01.004

[7] Hu, J.; Kaparias, I.; Bell, M.G.H.: Spatial econometrics models for congestion prediction with in-vehicle route guidance, Intelligent Transport Systems, 3(2), June 2009, 159–167. DOI:10.1049/iet-its:20070062

[8] Kaparias, I.; Bell, M.G.H.: Testing a reliable in-vehicle navigation algorithm in the field, Intelligent Transport Systems, 3(3), September 2009, 314–324. DOI:10.1049/iet-its2008.0075

[9] Sud, A.; Andersen, E.; Curtis, S.; Lin, M. C.: D Manocha, Real-Time Path Planning in Dynamic Virtual Environments Using Multiagent Navigation Graphs, IEEE Transaction on Visualization and Computer Graphics, May/June 2008, 14(3), 526-538. DOI:10.1109/TVCG.2008.27

[10] Aurenhammer, F.: Voronoi diagrams – a survey of a fundamental geometric data structure, ACM Computing Surveys (CSUR), 23(3), Sept. 1991. DOI:10.1145/116873.116880