Improving Real-Time Route Suggestions in Automotive Navigation Systems using Vehicle Cluster Behavior

S Rajarajeswari
Assistant Professor
Dept. of Computer Science
MSRIT, Bangalore, India

Sai Ram Prasad Reddy S
Dept. of Computer Science
MSRIT, Bangalore, India

Adithya Venkatesh
Dept. of Computer Science
MSRIT, Bangalore, India

ABSTRACT

Automotive navigation systems are widely used by drivers to help them reach a particular destination. Mostly, such systems are designed to suggest routes which will help the driver to reach destination by covering less distance or consuming less time. This paper presents a technique that will select least congested path by using a vehicular traffic prediction technique that utilizes Global positioning data paired with Speed and Accelerometer Telemetry provided by a GPS unit, mounted on a vehicle, to improve the results of existing graph search methodologies currently being implemented to provide navigation data to users. Clustering of vehicles is done, based on similar positional and directional behavior. These clusters will have similar congestion levels. Traffic data identified through clustering is used to manipulate the path cost of the corresponding road, on an existing road network graph. It tries to improve real-time route suggestions by selecting less congested routes dynamically.

General Terms

Automotive Navigation System, Real-Time Route Suggestions, Traffic Decongestion.

Keywords

Clustering, Dynamic Updates, Path Cost, Traffic Factor, Route Select Algorithm.

1. INTRODUCTION

An automotive navigation system is a satellite navigation system designed for use in automobiles. These days such systems are shipped in every smart phone. It makes use of a GPS navigation device to obtain position data to trace the user on a road in the unit's map database. Using the road database, the unit can give directions to other locations along roads also in its database. Moreover, these systems can not only give precise driving directions, but also receive and display information on traffic congestion and suggest alternate routes. Such systems may use either TMC, which delivers coded traffic information using radio RDS, or by GPRS/3G data transmission via mobile phones. A navigation system that not only helps the user to reach the destination in shortest time, but also enhances the travel experience by avoiding congested roads is a crucial requisite in this fast moving world.

Throughout the years, there has been high interest among the research community and various corporations in designing automotive navigation systems with various goals as indicated in [1-3]. System described in [1] provides useful information, including directions and safety instructions to drivers of automobiles. Additionally, the system in [2] provides spoken instructions in real time as they are needed. This paper presents a navigation system which can provide real-time alerts to avoid congested roads thereby not only decreasing travel time, but also decongesting roads and improving travel experience.

In this navigation system, identification of clusters of vehicles based on their location and direction in which they are moving is performed. Speed information from the cluster is used to set path costs for every road. Based on path costs, routes to the specified destination are ranked and presented to a particular user based on his location. Section 2 contains description of various components of the system and information flow within the system. Section 3 contains a graphical model for roads, where each path has certain cost. An algorithm for identification of clusters is described in Section 4 of this paper. Section 5 defines two important quantities the system uses to assign costs to each path. Section 6 describes an equation to compute costs for paths. Section 7 presents an algorithm called Route Select which selects less congested routes to reach a specified destination for a particular vehicle.

2. SYSTEM COMPONENTS AND INFORMATION FLOW

The system contains a centralized database of maps which communicates with multiple remote navigation systems installed in automobiles. The navigation system in the automobile is assumed to have a Global Positioning System (GPS) and a GPRS/3G data transmission facility via mobile phones. It is also assumed that speed measurements from speedometer are available to this device. According to [4], a GPS can be used to accurately determine the location and orientation of the vehicle accurately.

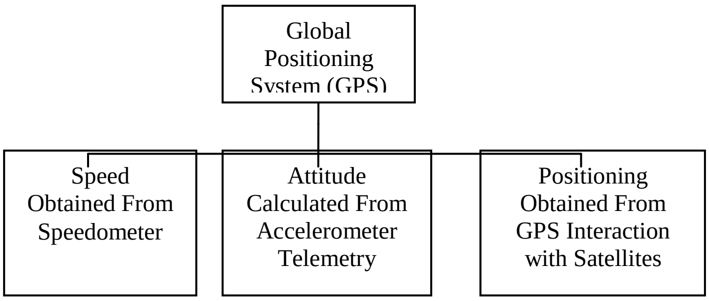

Fig 1: Client side representation of automotive navigation system

Information from GPS is used in the clustering algorithm described in section 3. The speed information transmitted from each vehicle is used to calculate average speed and moving average speed, described in section 5, that decide the path cost at regular intervals of time. The new path costs are assigned to roads in the maps of the database. Based on the vehicle's present location, congestion information about close by roads is sent to it. A pictorial representation of the system on the client side is shown in Figure 1. Figure 2 shows system architecture and information flow between system components.

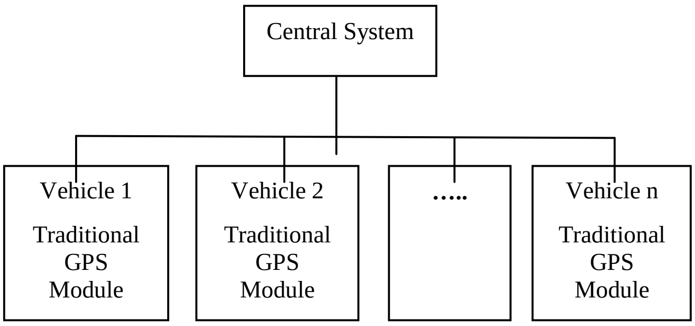

Fig 2: System architecture and Information Flow

3. A GRAPHICAL MODEL FOR ROAD NETWORKS

Graphs are frequently used in modeling roads and networks according to [7]. We use a directed-graph or digraph to model our road network. A digraph can be defined mathematically as follows according to [5-6]:

A digraph is an ordered pair of sets G = (V, A), where V is a set of vertices and A is a set of ordered pairs (called arcs) of vertices of V.

One useful variant of a digraph is called weighted digraph as described in [5-6]. A weighted digraph associates a label (weight or cost) with every edge in the digraph.

We model our road maps using weighted digraphs. Here, the vertex set V is the set of junction nodes where two or more roads intersect in the map or dead-ends where a road leads to nowhere. For every bi-directional road between the vertices V_i and V_j, we include the ordered pairs (V_i, V_j) and (V_j, V_i) in the edge set A. Here i and j are assumed to be some integer numbers. For every uni-directional road from vertex V_i to V_j, we include the ordered pair (V_i, V_j) in the edge set A. Weight or cost of each edge in the graph is computed to reflect congestion on that edge. An algorithm to assign costs to the edges is presented in Section 6.

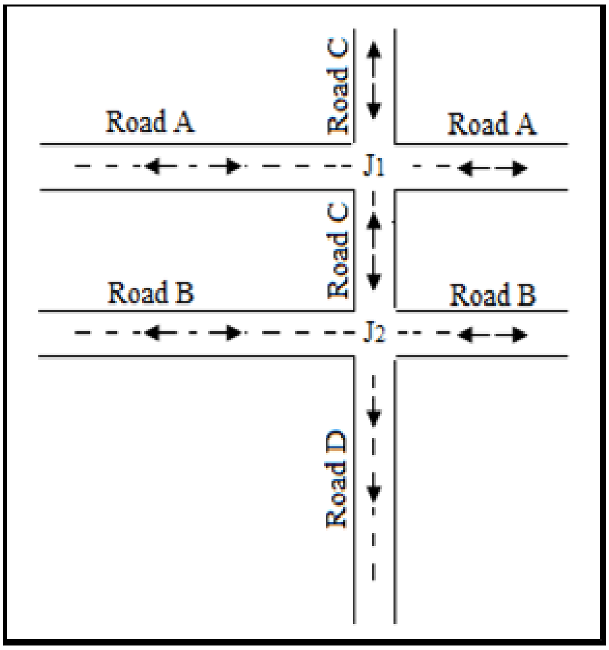

Fig 3: A sample road map with labels. Bi-directional Roads and Uni-directional roads are distinguished by the arrow directions on the map.

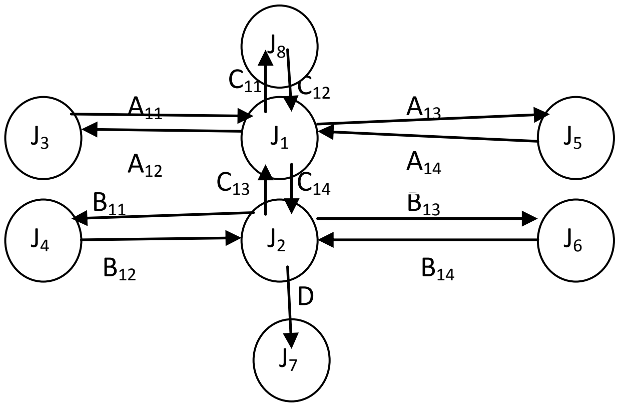

Fig 4: A digraph with nodes and directed edges drawn for road map shown in Fig 3.

4. CLUSTERING OF VEHICLES

Clustering of vehicles based on their location and direction of motion is performed. The central database server receives location, direction and speed from the client navigation system which is equipped with GPS and mobile connection. In each cluster only those vehicles that are moving in the same direction and are between the same junction points are included. Notice that each cluster will contain all vehicles that are on the same edge of the directed graph. This clustering helps us to calculate average speed, moving average speed and costs for any edge in set A of the graphical model described in Section 3.

The centralized map database server will cluster vehicles using the following algorithm:

4.1 Algorithm: Cluster

Inputs:

• L, GPS location coordinates sent by client.

• O, GPS orientation information sent by client.

Procedure:

1) Initialize C_i, the cluster set corresponding to the edge a_i in the edge set A in the graphical model, to null set.

2) Add vehicle V_i to C_i if L_i and O_i determine an edge a_i in the edge set A.

The algorithm above is based on the fact that current location and orientation will correspond to exactly one edge in the edge set A. For example, if vehicle V_i is travelling on the path represented by edge a_i in the system that connects V_i to V_i, then we add V_i to the cluster corresponding to edge a_i. Clustering in this manner will help the system have knowledge about traffic in all roads.

5. AVERAGE SPEED, EXPONENTIAL MOVING AVERAGE (E.M.A) SPEED OF A CLUSTER

In this section, a procedure to calculate two quantities that are useful in describing the present traffic on a particular path is presented. These quantities are the average speed of the cluster and exponential moving average (E.M.A.) speed of the cluster. Computation of average speed and E.M.A. is performed every second in order to provide reliable real-time updates. Note that speed is transmitted by the client navigation system to the central database server using a mobile data connection as described in section 2.

5.1 Average Speed

At time t, let the cluster C_i corresponding to directed edge a_i have n vehicles, $V_1, V_2,\ldots,V_n$, moving at speeds, $S_1, S_2,\ldots,S_n$, respectively. We define the average speed (M) of the cluster as follows:

$$M = (S_1 + S_2 + \ldots + S_n) / n \qquad (1)$$

Note that, at any point of time t, the average speed (M) calculated by equation (1) above reflects the present traffic condition on a particular path.

5.2 Exponential moving Average Speed

Average speed (M) calculated above alone cannot answer questions such as, “Is the path congested than normal?” or “Is the road deserted?” etc… To answer such questions we need to consider older information about traffic on the roads apart from information about current conditions provided by Average Speed (M). According to [8] an Exponential weighted moving average can be used to give importance to both present and old information. At any time t, let M_t be the average speed of the cluster, E_t be the current exponential weighted moving average to be computed. Let E_{t-1} be the exponential weighted moving average computed at time (t – 1). We define exponential moving average of the cluster E_t at time t as follows:

$$\text{If } t = 1, E_t = M_t \qquad (2)$$

$$\text{Else } E_t = \alpha.\ M_t + (1 - \alpha).\ M_{t-1} \qquad (3)$$

$$\alpha = 0.1 \qquad (4)$$

In equation (3) we use a constant α. Value of α assigns weight-age to new information and historical data. Higher value of α implies that more importance is being given to current information. In (4), we set α = 0.1 to give more importance to historical data. The E.M.A. speed calculated above provides information about normal traffic conditions on the road giving most importance to recent conditions. Note that, E.M.A. adapts very well to constantly varying conditions on roads.

6. COST ASSIGNMENT TO PATHS

On one hand, average speed (M) of the cluster represents the current traffic conditions. On the other hand, exponential moving average speed represents past traffic conditions. In this section, a term called Traffic Factor (F) is defined which tells us whether the current path is congested, normal or deserted. Next, traffic factor is transformed to make it more suitable to represent path costs.

6.1 Traffic Factor

At any point of time t, let the cluster C_i corresponding to directed edge a_i have average speed M_t and E.M.A speed E_t. Traffic Factor (F) is defined as follows:

$$F = M_t / E_t \qquad (5)$$

We can say that, if $0.5 < F < 1.5$, then the congestion on road corresponding to a_i is normal. If F is greater than 1.5, then the road is not congested. If it is less than 0.5, then the road is congested. Hence Traffic Factor (F) clearly tells whether the traffic is greater or lesser than normal.

6.2 Path Costs

Since Traffic Factor (F) can take fractional values such as 0.25 or 1.89, it is not so convenient to assign its value to edge costs. We prefer edge costs to be integer values such as 100, 256 etc… Here, an equation to transform traffic factor (F) to path cost (W) is shown.

$$W = \text{floor}\ (F * 100) \qquad (6)$$

Here, floor ( ) represents the greatest integer function. This way we can compute edge cost W_i for every edge a_i in the edge set A of our model described in Section 3.

7. ROOT SELECT ALGORITHM

So far, a model for road maps using weighted digraphs has been developed in this paper. Moreover, a method to assign costs to each edge in the graph was also presented. In this section, an algorithm called Route Select, that dynamically selects the next road, which when taken by a vehicle makes it reach the specified destination with least congestion while travelling, is introduced.

7.1 Background

In this navigation system, the user specifies the destination he wants to reach. The user's location, which was obtained using GPS, and the final destination are sent to the central server as described in Section 2. The source of the trip is always the current location being sent by the GPS. The source and destination can be any two random points accessible by roads on a map. The graphical model presented in section 3 has nodes corresponding to junctions and dead-ends. It completely ignores other points along the roads in the map. In order to find least cost path from source to destination using our graphical model we need to find approximate source and destination nodes.

7.2 Selecting source and destination nodes

Here, we present a way to approximate source and destination nodes for given source and destination points on the map. The destination point is fixed. Select a node closest by distance on the map to the destination point specified. Similarly, dynamically select the closest node by distance on the map to the current location. This is approximately the source node. Now, our goal is to reach the destination node from the source node with minimal congestion by dynamically selecting legs of the route.

7.3 Need to use A* search Algorithm differently

To make sure that each route-leg the system suggests to the user based on source and destination nodes will actually lead to the destination, the route-leg suggested should be in the path between source and destination. So, system needs to find paths between source and destination nodes.

If the path costs of our graph model are fixed and known in priori we could have used several Heuristic Search Algorithms described in [9-12] to find a path to the destination in one go. Using A* search Algorithm presented in [9] by Hart, one can ensure that the path has minimal cost.

However, one cannot use any Heuristic search strategy directly because the costs of edges in the graph are updated every second and the suggestions which the central system should provide are dynamic.

7.4 Route Select algorithm

Here, an Algorithm that generates dynamic route suggestions which will take the user from source node to destination node with minimal traffic congestion is presented. Whenever the vehicle is about to approach a junction, system needs to suggest which route-leg to take. System will apply A* search algorithm to find a minimum cost path to the destination node from this junction node at that point of time. It selects only the first route-leg from the path returned by the A* search. The system expects the user to travel along this route-leg towards the next junction. When vehicle approaches the next junction, system calls A* search algorithm again to find the current minimum cost path to the destination from the current location. Again, system selects the first route-leg from the path returned. It keeps doing this till the user reaches destination. This will ensure that the suggestions provided by the algorithm will adapt to the changing costs in the graph over time and suggest routes dynamically avoiding congested routes. Pseudo code for Route Select Algorithm is Figure 5.

We used the following notations in the algorithm specified in figure 5. The function Closest_Node ( ) takes map coordinates as parameter and returns closest graph node to it as specifies in sub-section B of this section. The function Edge ( ) takes current location and orientation information as inputs and returns the unique edge determined by the pair according to model described in section 3. The function Astarsearch ( ) takes source and destination nodes as parameters and returns least cost path between source and destination nodes. The function firstleg ( ) takes a path as a parameter and returns the first edge in the path. The function suggest ( ) takes an edge as a parameter and provides a suggestion to the client to take route corresponding to that edge.

```
Algorithm: Route Select

Inputs:
• L, location coordinates sent by client using GPS.
• O, Orientation information sent by client using GPS.
• D, Coordinates of destination sent by client.

Procedure:
d = Closest_Node (D);
Repeat
    e = Edge (L, O);
    s = v, where (u, v) represents edge e;
    p = firstleg (AstarSearch (s, d));
    suggest (p);
Until (Vehicle reaches destination D)
```

Fig 5: System architecture and Information Flow

8. CONCLUSION

In this paper, we have taken the audacious task of improving real-time route suggestions to reach a destination along least congested path. To accomplish the task, system architecture to process data was presented. Moreover, modeling of road maps using graphs in order to identify cluster of vehicles travelling between two adjacent nodes and in the same direction was done. A series of computations that helped us assign costs to the edges in the graph were presented. Finally, an algorithm that provides real-time dynamic suggestions to ensure that the vehicle reaches destination in least congested path was proposed.

9. REFERENCES

[1] Van Ryzin, "Automobile Navigation System," in United States Patent, No: 5,844,505.

[2] Davis, "Automobile Navigation System Using Real Time Spoken Driving Instructions," in United States Patent, No: 5,117,685.

[3] Golding, "Automobile Navigation System with Dynamic Traffic Data," in United States Patent, No: 5,933,100.

[4] Hofman-Wellenhof, Lichtenegger, Collins, “Global Positioning System. Theory and Practice,” Springer, Wien (Austria), 1993, ISBN 3-211-82477-4.

[5] Rosen, “Discrete Mathematics and its Applications,” McGraw-Hill, ISBN 978-0-07-338309-5.

[6] Bondy and Murty, “Graph Theory,” Springer, ISBN 0-486-24775-9.

[7] Bondy and Murty, “Graph Theory with Applications,” North-Holland, ISBN-10: 0444194517.

[8] Chou, “Statistical Analysis”, Holt International, 1975, ISBN 0-03-089422-0, Section 17.2.

[9] Hart, Nilsson and Raphael, “A formal basis for the heuristic determination of minimum cost paths“, in IEEE Transactions on Systems Science and Cybernetics”, July 1968.

[10] Pohl, “Heuristic search viewed as path finding in a graph”, in Artificial Intelligence 1 (1970), 193-204.

[11] Newell and Ernst, “The search for generality”, in Proc. IFIP Congress, 1965.

[12] Russell and Norvig, “Artificial Intelligence: A Modern Approach”, Pearson, ISBN: 978-81-775-8367-0.