

The Intelligent Travel Assistant

John F. DILLENBURG, Ouri WOLFSON and Peter C. NELSON

Abstract— The ultimate goal of the Intelligent Travel Assistant (ITA) is the fusion and development of a number of technologies into a device that will increase the efficiency of our transportation network through increased use of mass transit and ride sharing. The ITA will include: a) dynamic ridesharing, b) spatio-temporal database management, c) wireless communications, and d) a management framework.

The ITA itself is envisioned to be a compact portable device with the capability to plan multi-modal routes for its user. The traveler will enter a desired destination into the ITA and it will formulate several plans to get the user to this destination. The ITA will make use of wireless Internet technology to send and receive traffic information, transit schedules, and arrange for payment of fares. The Global Positioning System (GPS) will be used to track the user's current position for use in route planning and also as a means of assessing traffic conditions while en-route. A spatio-temporal database management system will be used to efficiently track ITA positions in real-time.

Index Terms—database systems, ride sharing, mobile communication, transportation

I. INTRODUCTION

THE ultimate goal of the Intelligent Travel Assistant (ITA) is the fusion and development of a number of technologies into a device that will increase the efficiency of our transportation network through increased use of mass transit and ride sharing.

Before looking into exactly what the Intelligent Travel Assistant concept is, a brief look into the reasons why such a device will be necessary is in order. The driving force for this research is the reduction of congestion on the nation's crowded roadway systems. Excess congestion cost the United States economy over \$78 billion in 1999 from fuel and wages alone [1].

Growth in the amount of automobile travel has increased over the past two decades by 76% [2]. Meanwhile, the amount of pavement to accommodate this travel has only increased by a meager 2% in the same period of time. The overall effect of

these two factors is evident in the rise in traffic congestion. The average annual delay due to traffic congestion has climbed over 300% in the past two decades, going from 11 hours stuck in traffic per person per year in 1980 to over 36 hours in 1999 [2]. Clearly, something must be done to reduce the amount of traffic, either by increasing the average number of occupants per vehicle through the use of car and van pooling, or through other mass transportation options.

Adding more roadways does not appear to be a viable option, either. The 2001 Urban Mobility Report from the Texas Transportation Institute shows that only half of the roads necessary to compensate for the increases in congestion were added between 1982 and 1999 [1]. There is not enough space, money and public approval for all the roads that would be needed to alleviate congestion. The American public does not necessarily want new roads either. A recent article in U.S. News and World Report [3] points to a survey done by Smart Growth America in which just 21% of the people surveyed believed adding new roads was the best long-term way of reducing congestion.

Over time, rising fuel prices further contribute to the losses incurred due to congestion. Overall, the annual cost of congestion has risen in step with congestion itself over the past two decades. In 1999, congestion cost over \$78 billion in the largest 68 urban areas of the United States [1]. That is over \$900 per person per year for the largest of these urban areas.

A number of steps can be taken to reduce the mobility problems that are and will continue to be a problem in the decade to come. Increases in vehicle occupants alone can make a significant impact on the levels of congestion. For instance, an increase of just 4% in the number of vehicles with multiple passengers would have been enough to offset the 1999 increase in congestion in the nation's largest 68 urban areas.

Although there is no one single answer to reducing congestion, increasing the efficiency of the transportation network through ride sharing is one of the most promising approaches. Unfortunately, not many large urban areas have comprehensive ride sharing programs available. One goal of the ITA will be to derive a standardized nation-wide system for communicating traveler information, including ride-sharing data, to the public.

Any ride sharing solution must be oriented to the end-user. Making it easier for the public at large to take advantage of potential ride share partners and mass transit will realize great improvements in congestion.

Manuscript received March 15, 2002. This work was supported in part by the Illinois Department of Transportation.

John F. Dillenburg is with the University of Illinois at Chicago Computer Science Department, Chicago, IL 60607-7053 USA (telephone: 312-996-5598, e-mail: dillengu@uic.edu).

Ouri Wolfson is with the University of Illinois at Chicago Computer Science Department, Chicago, IL 60607-7053 USA (telephone: 312-996-6770, e-mail: wolfson@uic.edu).

Peter C. Nelson is with the University of Illinois at Chicago Computer Science Department, Chicago, IL 60607-7053 USA (telephone: 312-996-3259, e-mail: nelson@uic.edu).

II. INTELLIGENT TRAVEL ASSISTANT

Although the ITA's ultimate form is not known, it will need the following features to be useful:

- Dynamic ridesharing: the ability to find ride share partners "on-the-fly"
- Multi-modal route planner
- Provide 24 hour traffic information:
 - Congestion
 - Construction
 - Incidents
 - Special events
- Convenient and easy to use
- Small and lightweight
- Built-in cell phone

A preliminary architecture for the ITA is shown in Figure 1. As can be seen, the ITA will depend on the global positioning system for determination of the user's current position. It will also need to be connected via a wireless Internet connection to a central traffic information computer from which it can obtain data on traffic congestion, incidents, construction, train schedules, bus routes, and other local ride share partners.

A. Requirements

A preliminary requirements analysis of the ITA is useful in determining the infrastructure requirements of the end product. Note that further requirement analysis will also be done as part of focus groups with relevant government transportation agencies and surveys of the traveling public.

1) User Input

The ITA will require input from the user in order to obtain the best possible route.

- Current Location

If the global positioning system is not functioning, the system will need the ability to "geocode" the user current position given a street address, intersection, mile-marker or landmark.

- Desired Destination

The desired destination can be entered from a list of common destinations such as "Home" or "Work" or can be geocoded from an address, intersection, mile-marker or landmark.

- Ranking Criterion for Plans

The ITA will need some idea of which routes are more desirable to the traveler than others so that they may be displayed in order of preference. Among the variables that may be used to rank routes are: estimated travel time, number of transfers, transportation mode, number of occupants if ride sharing, and smoking limitations.

- Desired Cost Limitations

Plans may also be filtered and rejected if they do not fit certain criteria, one of which would be the cost of the trip in question.

2) Location Database

A spatio-temporal database infrastructure will be needed to track the positions of all ITA devices, trains, buses, and

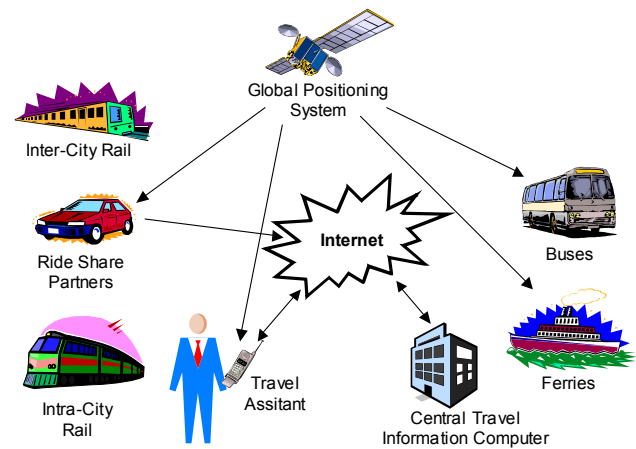


Figure 1. Intelligent Travel Assistant Concept

ferries. The database will need to support queries of the form "find all conveyances within a 1 mile radius."

3) Electronic Payment

The ITA will need the ability to negotiate and pay for travel. This payment should take place in a manner similar to the use of "e-cash" on the Internet.

4) Route Planning

The route planner portion of the ITA will be one of the most critical aspects of the system. It must be able to quickly and accurately find a route from the traveler's current position to the desired destination while taking into account the traveler's preferences, multiple modes of transportation, current traffic conditions, and possible future traffic conditions. The need for the latter is necessary given the fact that the traveler will be using the route in the future and so current traffic conditions are only an estimate of what they will be once the trip is completed. This implies that either the ITA itself or the Central Travel Computer will need the ability to forecast traffic conditions.

The route planner will also be needed for dynamic rerouting of drivers. The ITA will need to continually update the time and cost of the remaining trip by computing the route plan periodically. If the new route is enough of an improvement, it may be offered to the driver as an alternative to the existing route [5].

5) Ride Sharing Information

One way of increasing the average number of occupants per vehicle is through ride sharing. The ITA will need the ability to offer ride share services in two modes: pre-determined and dynamically. Pre-determined ride shares are those which are arranged ahead of time and which take place between fixed locations such as home and work. Dynamic ride sharing was illustrated in the travel scenario in section C.2. Basically, dynamic ride sharing allows the system to track ride share drivers along their travel route and allows additional riders to be picked up en-route.

For dynamic ride-sharing to be feasible, the current position of the ride share driver will need to be tracked. This may be accomplished by the driver's ITA transmitting its current

position to the central travel computer periodically. The route and travel schedule of the driver can also be used to enhance accuracy, reduce the amount of data transmitted [8], or as a back-up means of determining the driver's location in the case of communication failures.

Dynamic ride sharing will also require a means of determining the status of drivers, such as the current number of occupants and the preferences of the driver (how far off-route he is willing to travel, what fees will be charged, etc). Alternatively, or in addition to these, the driver will be given notification of a potential ride share partner and be given the choice of whether to pick the partner up.

6) Transit Information

In addition to ride sharing, mass transit provides another means of increasing the efficiency of the transportation network. To make use of mass transit facilities, the ITA will need the schedules and routes of all buses, trains and ferries. For the location of our testbed, the Chicago Metropolitan area, these schedules are currently available on the CTA website, www.transitchicago.com, and the RTA website, www.metrarail.com.

7) Traffic Conditions

Current traffic conditions will need to be collected and distributed to the ITA so that it may determine the fastest route to a destination. Congestion, construction, incidents and special events (parades, sports, etc) are the most critical traffic conditions the ITA will need. There are currently in place a number of traffic information clearinghouses on the Internet that can be used for this purpose, such as www.travelinfo.org and www.traffic.com. For our testbed area, we will use the Gateway system we have developed for the Illinois Department of Transportation as the primary source for real-time traffic data. Online traffic information is also an area currently under development and for which standards are needed for the format and content of the information to be distributed.

8) Telephony

An integrated telephone will make negotiations with a potential ride-share partner easier in some cases and will also allow reservations to be made at the destination if necessary. Some examples of devices on the market today with the capability of providing telephone, GPS and Internet are the Handspring Visor and the Palm i705.

9) Global Positioning System

A global positioning system (GPS) receiver will be needed in order for the ITA to determine the traveler's current position and also to send periodic updates of the traveler's position to the central travel computer for use in ride sharing scenarios.

B. Design Elements

1) Location Management Database

Location management provides a mechanism for querying a database for the current location of ITA devices, buses, trains, and other equipped vehicles. Queries will often be set oriented, locations of finer resolution are necessary, queries may pertain

to the future as well as the past, and triggers may be more important than queries. Some examples of queries/triggers are: notify me when a conveyance is within 1 mile of my projected location in 5 minutes (trigger); how many times was bus #5 late by more than 10 minutes at station 12 (past query).

a) A Naïve Solution and Its Drawbacks

A fundamental capability of location management is modeling of transient location information, particularly the location of mobile devices such as cell phones, personal digital assistants, laptops, etc. These devices are carried by people, or mounted on moving objects such as vehicles, aircraft, or vessels. The location information is updated by positioning technologies.

A straightforward approach that is used by existing applications is to model the location as follows. For each moving object, a location-time point of the form (l, t) is generated periodically, indicating that the object is at location l at time t . The variable l may be a coordinate pair (x,y) , or a *cell-id*. The point is stored in a database managed by a Database Management System (DBMS), and SQL is used to retrieve the location information.

This method is called point-location management, and it has several critical drawbacks. First, the method does not enable interpolation or extrapolation. For example, information can only be retrieved for the moving objects that happened to generate a location update at the requested time. The problem is even more severe for extrapolation, i.e. if a future location is requested This query cannot be answered by the point-location method.

The second problem of the point-location method is that it leads to a critical precision/resource trade-off. An accurate picture of the precise location of moving objects would require frequent location updates that consume precious resources such as bandwidth and processing power.

Finally, a third problem of this method is that it leads to cumbersome and inefficient software development. Specifically, location based services will require the development of a vast array of new software applications. Doing so on top of existing DBMS technology has several drawbacks. First, existing DBMS's are not well equipped to handle continuously changing data, such as the location of moving objects. The second drawback is that location based applications need to manage space and time information, whereas SQL is not designed and optimized for these types of queries and triggers. Finally, the location of a moving object is inherently imprecise because the database location of the object (i.e. the object-location stored in the database) cannot always be identical to the actual location of the object. This inherent uncertainty has various implications for database modeling, querying, and indexing. For example, there can be two different kinds of answers to queries, i.e. the set of objects that "may" satisfy the query, and the set that "must" satisfy the query. SQL semantics cannot account for this difference.

b) *Trajectory Location Management*

In this section we outline our proposed model of a trajectory, explain how to construct it, and explain how it solves the problems associated with point location management. Let us observe that there are alternatives to the approach here (see for example [29, 30]). If possible, we make use of a-priori or inferred information about the destination of an object.

The method proposed is called trajectory location management. In this method, we first obtain or estimate the source and destination of the moving object. For example, the object starts in New York City at 57th street at 8th Ave. at 7am and heads for Chicago to the intersection of Oak and State streets. Then, by using an electronic map geo-coded with distance and travel-time information for every road section, a trajectory is constructed.

Before defining the trajectory, let us define the format of an electronic map. An electronic map is a relation. Each tuple in the relation represents a city block, i.e. the road section in between two intersections, with the following attributes:

- Polyline: the block polyline given by a sequence of 2D x,y coordinates: $(x_1,y_1),(x_2,y_2),\dots,(x_n,y_n)$.
- Fid: The block id

The following attributes are used for computing travel-time and travel-distance.

- *Meters*: length of the block in meters
- *Drive Time*: typical drive time from one end of the block to the other, in minutes

Thus each map is an undirected graph, with the tuples representing edges of the graph.

The route of a moving object O is specified by giving the starting address or (x,y) coordinate (*start_point*), the starting time, and the ending address or (x,y) coordinate (*end_point*). An external routine computes the shortest cost (distance or travel-time) path in the map graph. This path, denoted $P(O)$, is given as a sequence of blocks. Since $P(O)$ is a path in the map, the endpoint of one block polyline is the beginning point of the next block polyline. Thus the whole route represented by $P(O)$ is a polyline denoted $L(O)$.

Given that the trip has a starting time, for each straight line segment on $L(O)$, we can compute the time at which the object O will arrive to the point at the beginning of the segment (using the *Drive-Time* attribute). This is the certain-trajectory, or c-trajectory. The c-trajectory is a sequence of straight-line segments $(x_1, y_1, t_1), (x_2, y_2, t_2), \dots, (x_n, y_n, t_n)$ in 3-dimensional space. The c-trajectory means that when the object starts at a location having coordinates (x_1, y_1) at time t_1 it will move on a straight line at constant speed and will reach location (x_2, y_2) at time t_2 , and then it will move on a straight line at constant speed and will reach location (x_3, y_3) at time t_3 , etc. The c-trajectory is an approximation of the expected motion of the object in space and time. The number of line segments on the trajectory has an important implication on the performance and precision of queries and triggers. Specifically, the performance increases and the precision

decreases as the number of line segments decreases. We adjust and fine-tune the number of line segments on each trajectory by using a method that has been studied in computer graphics, namely line simplification (see [18], [19]).

The c-trajectory is stored in the server database and in a computer on board the moving object. At any point in time t between t_i and t_{i+1} the server can compute the expected location of the moving object at time t . Observe that this technique solves our first problem by enabling both location interpolation and extrapolation. The server can compute the expected location of the moving object at any point in time between the start and end times of the trip.

Finally, the trajectory is obtained by associating an uncertainty threshold u_i with the i^{th} line segment on the c-trajectory. The line segment together with the uncertainty threshold constitute an "agreement" between the moving object and the server. The moving object will update the server if and only if it deviates from its expected location according to the trajectory by u_i or more. The moving object can obtain the deviation at any point in time by comparing a GPS update with its computed location. The deviation is simply the distance between the actual and the expected location.

At the server, the trajectory is maintained by revising it according to location-updates from the moving object, and according to real-time traffic conditions obtained from traffic web sites. We have developed a traffic incident model, and a method of identifying the trajectories affected by a traffic incident. Observe that determining whether or not a trajectory is affected by a traffic incident is not a simple matter, and it requires prediction capabilities. We use historical information and a novel traffic model to make this prediction.

Observe that the agreement between the moving object and the server solves the second problem of point location management. Namely, the tradeoff between resource/bandwidth consumption and precision has been broken. In trajectory location management the location of a moving object can be computed with a high degree of precision, using a small number of location updates, or no updates at all. In particular, if the moving object is "on schedule," i.e., it does not deviate from its prescribed trajectory by more than the uncertainty threshold, then no resources are consumed for updates.

Finally, let us observe that a trajectory can be constructed based on past motion in which an object used the point location management. Namely, the trajectory can be constructed from a set of 3D points $(x_1, y_1, t_1), (x_2, y_2, t_2), \dots, (x_n, y_n, t_n)$ that were transmitted by a moving object using the point location management method. One can simply connect the points along the shortest path on the map, and then associate an uncertainty u_i with line segment i . The uncertainty u_i can be bounded given the maximum speed of the object and the known times of the two GPS points immediately preceding and succeeding the i^{th} line segment (see [14]).

c) *Data Access Operators*

Finally, we propose to solve the third problem associated with point location management using a novel set of operators by which the database is accessed. The operators are used to query the database and also to set triggers that are fired when interesting conditions are satisfied by the database. The operators are designed to express when/where questions in an uncertain environment. This means that one can ask queries and set triggers that combine the traditional database conditions with the new operators. This also means that the operators can be combined using boolean operators such as AND and OR. An additional implication is that a user can enter these operators/queries on a client computer and the same set of operators can be invoked from a program. The latter option enables development of complex spatial and temporal applications such as the ITA.

The new operators are divided into three classes, operators that pertain to a single trajectory, operators that pertain to the relationship of trajectories to fixed-location facilities or regions, and the relationship among multiple trajectories. These loosely correspond to point queries, range queries, and join queries, respectively, in traditional databases (see [20]). Each one of the immediately following subsections discusses one of these classes.

Some operators that analyze a single trajectory.

1. **WHEN object o CLOSEST TO address x .** The operator returns a list of times at which the object passes by or stops at address x . Observe that there may be a list of times since the object may visit or pass by the same location more than once. If the object never passes by or visits x , then the operator returns the time when the object passes by the closest location to x on its trajectory.
2. **VCR object o .** This operator "replays" the trajectory of object o . The replay can be done on a certain time-scale (e.g. a minute per second), and it can fast forward or rewind to a certain point in time.

Some operators for retrieving trajectories that stand in certain relationships to a region or a facility.

Each one of the operators in this class is a condition. The condition is satisfied by the objects that stand in a certain relationship (e.g. within distance x) to a fixed facility (i.e. a point on a map) or a region R , during T . Thus, the conditions correspond to a spatio-temporal range query.

Some operators for identifying relationships between trajectories.

Each one of the relationship-to-facilities operators can be applied as a relationship-between-trajectories operator. These are called join operators. For example:

Possibly-Within [distance d | travel-time t], Sometime in the time interval T . The condition is satisfied by the pairs of trajectories that are within distance d or travel time t from each other, sometime in the time interval T . This operator is used, for example, in an air-traffic-control system that stores the trajectories of planes.

The opposite operator also applies. Specifically, **Possibly-**

Farther than [distance d | travel-time t], Sometime in the time interval T .

d) *Uncertainty Management*

The location of a moving object is inherently imprecise due to motion and unreliable location computations. Therefore, the database location cannot always be identical to the actual location of the object. Systems that do not manage this uncertainty delegate to the user the responsibility of understanding and taking into consideration its implications. The objective of uncertainty management is to assist the user in this task. This objective has various implications for database modeling (in our model the uncertainty is part of the trajectory), querying (possibly and definitely operators), indexing, and resource consumption.

Assuming that one can control the amount of uncertainty in the system, how should it be determined? Obviously, lowering the uncertainty would come at a cost. For example, if a moving object transmits its location to a location-database every x minutes or every x miles, then lowering x would decrease the uncertainty in the system, but increase bandwidth consumption and location-update processing cost; and vice versa, increasing x would increase the uncertainty but decrease resource consumption. Similarly, adjusting the uncertainty thresholds u_i in our trajectory model has the same tradeoffs concerning resource consumption.

Next we outline our cost based approach to quantify this tradeoff as a demonstration of a possible formalization of the problem (see [15]). The information cost of a trip has the following three components: deviation cost, uncertainty cost, and communication cost. Using these costs we define a function that represents the overall information cost of a trip, and define the optimal uncertainty threshold as the value that minimizes this function.

We believe that the method of defining an uncertainty threshold and communicating only values that exceed the threshold is an important paradigm that has applications beyond location management. Indeed, the uncertainty threshold paradigm was used in the context of data warehousing (see [16]) and general sensors (see [17]).

e) *Location Prediction*

Our proposed system does not always have a priori information about the future motion of a user. In other words, in contrast to other enterprise systems discussed so far, a user does not need to provide her destination. Location prediction is important in other applications such as wireless bandwidth allocation (in a cellular architecture, location prediction enables optimizing allocation of bandwidth to cells).

We have developed methods of motion prediction based on historical trajectories of moving objects. Our prediction methodology is based on the fact that often moving objects have some degree of regularity in their motion. That is, motion has a random part and a regular part, and the regular part has a periodic pattern (hourly, weekly, etc.). Therefore, we

decompose the motion prediction problem into two sub-problems: periodicity detection, and location prediction based on detected periodicity.

Periodicity detection seeks motion patterns. Assuming that we are given time-stamped sets of GPS points, the following are features of the patterns. First, patterns are partially periodic, i.e. sometimes only part of the motion repeats. For example, a person may usually travel from home to work along a fixed route between 7am and 8am every workday and back home between 5pm and 6pm. She may do other things and go other places during the rest of the day, and this constitutes the random part of the motion.

Second, the patterns are not necessarily repeated perfectly. For example, the home-to-work trajectory on one day may be different than on another day, due to different traffic conditions. Or, the person may decide to stay at home some workdays and thus miss certain periods. Finally, the motion can have multiple periodic cycles. For example, the person may go fishing every Saturday and every other Sunday.

In summary, the goal is to detect motion patterns that can be partially periodic, not perfectly repeated, and have multiple periodic cycles.

2) Route Planning

An efficient route planner that is capable of handling multi-modal routes and also makes use of real-time traffic information, traveler preferences, and cost limitations will need to be studied. Several artificial intelligence based algorithms could be modified for use with an intelligent travel assistant. Heuristic search algorithms such as A* [9] are particularly well suited for finding routes in road networks. Efficiency improvements such as the use of perimeters [6][7] can be used to insure routes are found in a timely manner. The ITA will most likely have tight memory constraints, so search algorithms designed to work with a fixed amount of memory will be needed [10][11]. The search can also be conducted on the server and the results forwarded to the ITA.

3) Traffic Prediction

The effectiveness of a route planner depends heavily on the accuracy, credibility and reliability of traffic prediction. Many artificial intelligence algorithms can be used to predict the traffic, with artificial neural networks (ANNs) showing the most promise.

Traffic flows reflect a series of underlying highly non-linear relationships. ANNs are notable for their use in addressing non-linear problems, an important trait when dealing with highly dynamic traffic data. ANN prediction offers several other benefits as well. With ANNs, freeway modeling is unnecessary. In addition, properly trained neural networks are relatively insensitive to erroneous or missing data. This is a valuable asset in traffic prediction since the input data is susceptible to noisy signals, transmission errors, and mechanical failure.

Several studies [12][13] have clearly demonstrated the feasibility of using neural networks for traffic prediction. These results indicate that neural networks are capable of predicting travel times up to 15 minutes into the future with a

high degree of accuracy (93% to 95%). Further improvements in model performance may be obtained using more complex neural network architectures and additional inputs from previous time intervals. For ANNs to be viable for on-line applications, they will need to be able to function in real time, processing real data. So training data must ultimately consist of real data that are automatically collected and processed.

REFERENCES

- [1] "The 2001 Urban Mobility Report," David Schrank and Tim Lomax, Texas Transportation Institute, <http://mobility.tamu.edu/>
- [2] "Pocket Guide to Transportation," Bureau of Transportation Statistics, <http://www.bts.gov/publications/pocketguide/pgtt00.pdf>
- [3] "American Gridlock," Phillip J. Longman, U.S. News and World Report, <http://www.usenews.com/usnews.issue.010528/usnews.traffic.htm>
- [4] "Broadband Wireless, Integrated Services, and Their Application to Intelligent Transportation Systems", Keith Biesecker, http://www.itsdocs.fhwa.dot.gov/jpodocs/repts_te/@5_01!.pdf
- [5] "Real-time Traffic, Rerouting Faces Some Uncertainty," Amy Gilroy, Tvinsite.com, http://www.tvinsite.com/twice/index.asp?layout=story_stocks&articleid=CA83081&display=Mobile+Electronics&title=Real%2Dtime+Traffic%20C+Rerouting+Faces+Some+Uncertainty
- [6] "Techniques for Improving the Efficiency of Heuristic Search," John F. Dillenburg, Ph.D. Dissertation, University of Illinois at Chicago, Chicago, IL (1993)
- [7] "Perimeter Search," John F. Dillenburg and Peter C. Nelson, Artificial Intelligence 65, 165-178 (1994)
- [8] "Modeling and Querying Moving Objects," P. Sistla, O. Wolfson, S. Chamberlain, S. Dao, Proceedings of the Thirteenth International Conference on Data Engineering (ICDE13), Birmingham, UK, Apr. 1997
- [9] "Principles of Artificial Intelligence," Nils J. Nilsson, Los Altos, Morgan and Kaufmann (1980)
- [10] "Fast Recursive Formulations for Best-First Search that Allow for Controlled Use of Memory", A. Sen and A. Bagchi, Proceedings of the 11th International Joint Conference of Artificial Intelligence (IJCAI-89), 297-302 (1989)
- [11] "Heuristic Search in Restricted Memory", P.P. Chakrabarti, S. Ghose, A. Acharaya, and S.C. De Sarkar, Artificial Intelligence 41, 197-221 (1989)
- [12] "Freeway travel time estimation using neural networks," Hussein Dia, Proceedings of the ITSA'99 4th international conference on smart solutions at work, 1999
- [13] "Freeway traffic data prediction using neural networks," C. Taylor and D. Meldrum, Proceedings of IEEE Pacific Rim TransTech Conference, Seattle, Washington, August 1995
- [14] "Capturing the uncertainty of moving objects representations," D. Pfoser, C.S. Jensen, Proc. of the 12 Intl. Conf. on Scientific and Statistical Database Management, 2000. IEEE Computer Society
- [15] "Updating and Querying Databases that Track Mobile Units," O. Wolfson, A. P. Sistla, S. Chamberlain, Yelena Yesha Distributed and Parallel Databases, 7, 257-287, 1999
- [16] "Offering a Precision-Performance Tradeoff for Aggregation Queries over Replicated Data," C. Olston, J. Widom, Twenty-Sixth International Conference on Very Large Data Bases (VLDB 2000), Cairo, Egypt, September 2000
- [17] "Divergence Caching in Client-Server Architectures," Y. Huang, R. Sloan, O. Wolfson, Proceedings of the third International Conference on Parallel and Distributed Information Systems (PDIS), Austin, TX, Sept. 1994, pp. 131-139
- [18] "Efficient Algorithms for Approximating Polygonal Chains," P.K. Agarwal and K. R. Varadarajan.. Discrete Comput. Geom., 23:273-291(2000)
- [19] "Algorithms for the reduction of the number of points required to represent a digitized line or its caricature," D.H. Douglas and T. K. Peucker. Canad. Cartog. 10(2):112-122, Dec. 1973

- [20] "Database System Implementation," H. Garcia-Molina, J. D. Ullman, J. Widom., Prentice Hall, Upper Saddle River, NJ