Development of a Web-based Arterial Network Analysis System for Real-time Decision Support

Yao-Jan Wu (Corresponding Author)
Graduate Research Assistant
Department of Civil and Environmental Engineering
University of Washington
Seattle, WA 98195-2700
Tel: (206) 543-7827
Email: yaojan@u.washington.edu

Shi An, PhD
Professor
School of Transportation Science and Engineering
Harbin Institute of Technology
202 Haihe Road, Nangang District
Harbin 150090, P.R. China
Tel: +86 (451)8641-8577
Fax: +86 (451) 8628-3779
Email: anshi@hit.edu.cn

Xiaolei Ma
Graduate Research Assistant
Department of Civil and Environmental Engineering
University of Washington
Seattle, WA 98195-2700
Tel: (206) 543-7827
Email: xiaolm@u.washington.edu

Yinhai Wang, Ph.D.
Associate Professor
Department of Civil and Environmental Engineering
University of Washington
Seattle, WA 98195-2700
Tel: (206) 616-2696
Fax: (206) 543-1543
Email: yinhai@u.washington.edu

Word count: 5926 + 6 figures *250 (1500)= 7426 words
Submitted for Presentation at the 90th Annual Meeting of the Transportation Research Board and for Publication in Transportation Research Record: the Journal of the Transportation Research Board.
Resubmitted on Nov. 15, 2010

ABSTRACT

With increasing data collection for Intelligent Transportation System (ITS) for arterial networks, archiving, managing and analyzing complex network traffic data is becoming challenging. Challenges include inconsistent data connections, data quality control, query performance, traffic prediction, and computational limitations. In order to deal with these challenges, this paper presents a web-based Real-time Analysis and Decision-making for ARterial Network (RADAR Net) system. This system adopts a relational database that consists of link, intersection and detector entities. The relational data demonstrates its query performance and scalability. The system contains four layers: offline server, online server (middleware), online server (Java Servlet) and online client. This four-layer design successfully distributes the computational burden of the system. In order to monitor the arterial performance, link speeds are calculated directly from the loop detector data retrieved from the City of Bellevue, WA. The system can dynamically predict and smooth real-time loop spot speeds by using α–β filter, a simplified version of Kalman filter while maintaining high system performance. The link speeds of the entire network are calculated and updated in real-time. Based on the system architecture, many application modules, e.g. capacity analysis and dynamic routing, were implemented and proved the system feasible to perform real-time analysis and assist decision making.

Key words: Arterial Performance Measurement, Speed Estimation, Speed Prediction, Dynamic Filtering, and Decision Support System (DSS).

1. INTRODUCTION

With the new technology developments in Intelligent Transportation Systems (ITS), increased deployments of traffic sensing technologies can easily provide the large amounts of live traffic data necessary for real-time transportation management, e.g. incident detection, traffic operations, and performance measurement. An arterial management system is regarded as one the most challenging information system because it requires additional effort to clean, archive, analyze, and interpret the data describing complex traffic conditions. Raw data gathered from sophisticated sensor networks requires further processing to produce useful results for traffic management and traveler information systems. It has been a challenging issue to manage and utilize traffic sensor data effectively. For example, traffic sensor data must be processed before being transferred to Advanced Traveler Information Systems (ATIS). However, ATIS cannot be informative and successful without a well designed database and data analysis methods. In order to support real-time information display and historical data analysis, the idea of Archived Data User Service (ADUS) has been proposed since the 1990s, allowing transportation agencies to efficiently store and redistribute ITS-generated data for analysis *(1)*. With the improvement of information technology, web-based systems have become popular (e.g. *(2,3,4)*) because these systems can efficiently display, analyze and disseminate traffic information in a timely manner. Because of these advantages, this web-based system is suitable for supporting real-time decision making, such as emergency evacuation plan development and execution and emergency vehicle routing. However, at the current stage, most web-based ATIS and ADUS focus on freeway applications and few address issues on urban streets. For example, Chen (*3*) and Bertini et al. *(4)* respectively developed Freeway Performance Measurement System (PeMS) and Portland Regional Transportation Archive Listing (PORTAL), two major ATIS systems with ADUS for freeway applications. The Regional Integrated Transportation Information System by Pack et al. *(5)* is one of the most comprehensive ATIS and ADUS that demonstrates capabilities of region-wide automated data sharing, dissemination, and archiving. However, their effort for real-time arterial data processing was not addressed. Even though Arterial Performance Measurement System (APeMS) was recently developed by Petty et al. (*6*), the functionality was limited and not capable of analyzing a large-scale arterial network. Based on current state of the art, most system development has difficulties in providing real-time network-wide arterial analysis and decision support functions possibly due to the lack of real-time arterial data. Hence, techniques for real-time arterial data processing and analysis have not been fully explored by researchers. Nowadays, more and more cities (e.g. City of Bellevue, WA) are capable of providing high-resolution arterial traffic data. If the data can be timely processed, the real-time results can assist road users and engineers in real-time decision making in a complicated arterial network and, meanwhile, provides researchers with a foundation to solve theoretical network problems that have not been verified in the past.

The ideal real-time arterial network system for decision making has several requirements, including responsiveness to queries, system flexibility, scalability, and real-time computing. However, several prevailing challenges for such system are discussed below.

1. Inconsistent data connections

There are several ways to transmit real-time data between the data providers and clients. For example, Washington State Department of Transportation (WSDOT) adopts Simple Object Access Protocol (SOAP) to disseminate real time incident Extensible Markup Language (XML) data. The City of Bellevue, WA, archives traffic data as flat files in the data server and the public can fetch the data via File Transfer Protocol (FTP). Regardless of data transmission methods, the data could be missing while being transmitted from the on-site sensors to the Traffic Management Center (TMC). It is often observed in practice that communication fails periodically.

2. Data quality control

A data quality control procedure is a key to provide accurate results. Some erroneous data should be removed. For example, loop detectors generally have sensitivity errors *(7)*, resulting in wrong detection readings. Moreover, speed estimation should be corrected in the situations where occupancy or volume is zero. These erroneous data could be discarded; meanwhile, more data would be lost.

3. Query performance

Arterial networks usually contain hundreds of roadway links and intersections. With the improvement of ITS data collection infrastructure, huge amounts of data are transmitted to the data warehouse. The key to improving query performance is an efficient database design

4. Traffic prediction

The traffic status changes dynamically with some randomness. Short-term prediction has been a critical issue. Most prediction algorithms require high computational power and are not suitable to implement in a real-time system. Most decision making processes, e.g. shortest path routing estimation, require smoothing and prediction process for the detector measurements, e.g. volume or speeds. For a real-time decision support system, the performance of short term prediction should be taken into account. Even though the quality control procedure may discard most of the erroneous data, the impact of malfunctioning detectors and systematic errors should not be ignored. The prediction mechanism needs to be tolerant of noise and error in order to minimize the impact of erroneous data.

.

5. Computational limitation

Calculating statistics and algorithm implementation require computational power. If the computation burden is only on the server side, server performance will be impacted. Arterial networks usually have many links and nodes (intersections) with a large amount of data to process. Distributed computing can mitigate resource problems and should be considered in the system design.

To effectively analyze and disseminate the arterial network information to decision makers, traffic engineers and researchers, the main objective of this study is to overcome the aforementioned critical issues and develop a web-based Real-time Analysis and Decision-making for ARterial Network (RADAR Net) system. Therefore, the remainder of the paper is organized as follows. First, the database and system designs of RADAR Net will be introduced. Next, the details of the design flow will be elaborated upon followed by a detailed consideration of the loop spot speed estimation and prediction, and link speed calculation. The implementation of each functional module will be introduced and the system performance will be discussed. In this end, the paper will be concluded with lessons learned, recommendations and future work. The RADAR Net system is implemented as a sub-system of the Digital Roadway and Interactive

Visualization and Evaluation Network (DRIVE Net), an online interdisciplinary data integration and analysis platform, at www.uwdrive.net hosted by the Smart Transportation and Application Research Laboratory (STAR Lab) at the University of Washington.

2. FRAMEWORK AND SYSTEM DESIGN

2.1 Network Description

As of July 2010, the City of Bellevue, WA operates more than 182 signalized intersections, in which 165 signals are connected to a centralized computer system operated to archive all the traffic data in Bellevue's traffic management center (TMC). Real-time traffic data, e.g. volume and occupancy, is mainly retrieved from advance loop detectors located 100 ~ 130 feet (30.5 ~ 45.7 m) upstream from the stop bar of each approach. All the data are stored in a FTP data server and downloaded automatically into the DRIVE Net arterial database every minute. More details of the data retrieval process can be founded in *(8,9)*. The intersections with real-time data are displayed in the traffic light icons in Figure 1. As of July 15th 2010, real-time traffic data from 706 loop detectors are sent data back to Bellevue's TMC.

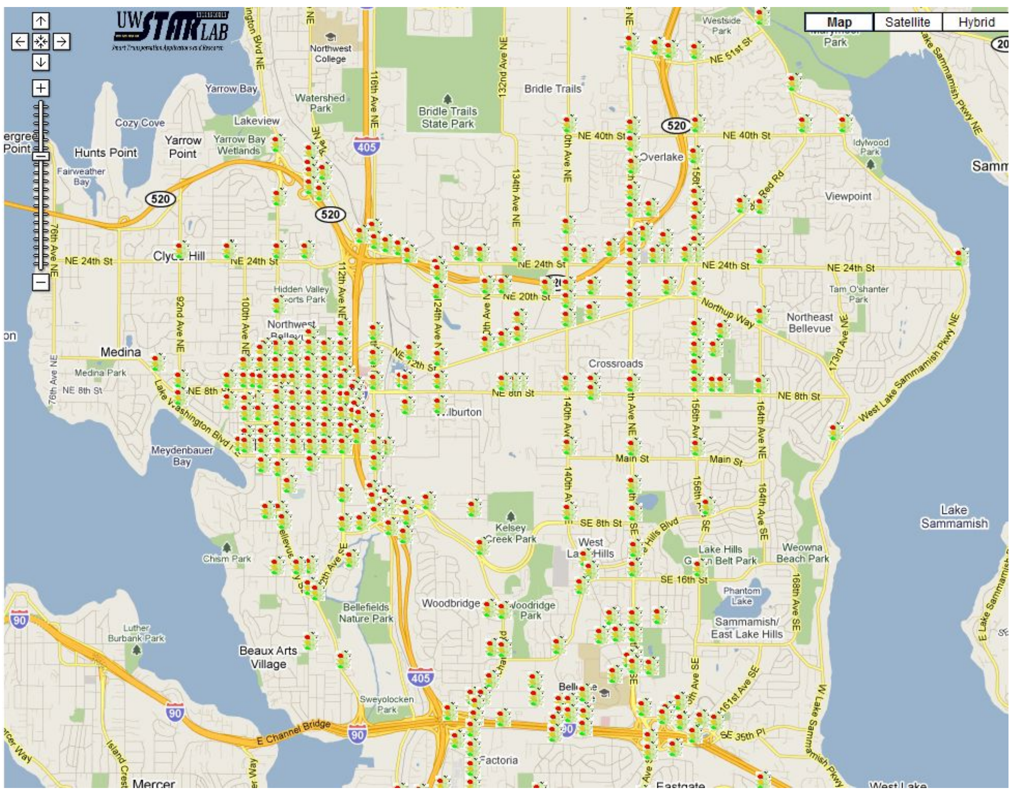


Figure 1: Study arterial network in the City of Bellevue, WA

2.2 Previous work

The Google-Map-based Arterial Traveler Information (GATI) system has been running since 2007 (*8,9*). The GATI system provides real-time traffic information, historical data query and two visualization functions, scatter and time-domain plots for volume and occupancy. The analytical statistics can be calculated online based on the users' inputs. However, this system suffers from several drawbacks.

- The GATI system is programmed in JavaScript and PHP. Few integrated development environment (IDE) software packages are designed for JavaScript and PHP. The debugging process tends to be slow and tedious.
- Fewer codes and libraries can be found and reused even though JavaScript and PHP are objected oriented. It is probably because of the low programmability (e.g. difficult to debug), few developers are willing to develop and share code .
- The visual component of GATI was hard coded in Cascading Style Sheets (CSS). The interface is difficult to adjust and fit to all types of browser settings.
- The visualization module was completed with third-party packages. The visualization flexibility is limited.
- The database has dependency issue. This issue increases the database size.

These issues are also commonly observed in practical applications. To deal with these issues mentioned above, the RADAR Net system aims to renovate the GATI system design using improved system and database designs.

2.3 Database Design

Real-time decision making relies on prompt query response from databases. For a typical On-Line Transaction Processing (OLTP) system, database design is a key to retrieve timely data through query. The relational database *(10)*, commonly used for OLTP systems, is used in RADAR Net. The relational database can provide many advantages. For example, the data are organized in different relations (tables) and use Structured Query Language (SQL) to query the specific results as desired *(11)*. Moreover, new relations and attributes can be easily added to the design and increase the design flexibility and database scalability.

The database design proposed in the previous research *(8)* was found to contain data dependency and anomalies. The redundant data occupied more than 40% storage space. Therefore, the Entity-Relationship (E/R) diagram was further improved, as shown in Figure 2.

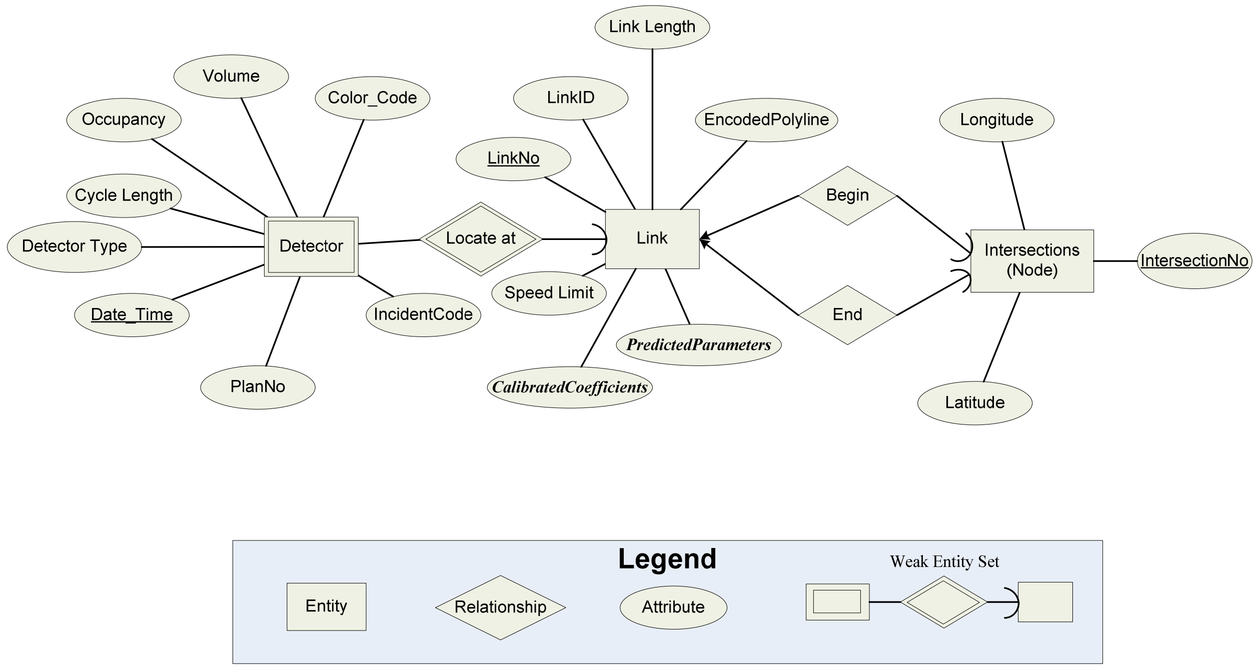


Figure 2: E/R diagram design for the arterial network database

Schemas

The E/R diagram in Figure 2 is converted to the schema following the conversion principle of the E/R data model *(11)*. These schemas represent three tables in the SQL database. The Detector table stores the real-time detector data. The link and Intersection tables store the time-independent attributes. Thus, users can add/update links or intersections without affecting the Detector table. The attributes are briefly explained as below.

1. Detector(<u>Date_Time, LinkNo</u>, Volume, Occupancy, PlanNo, Cycle Length, ColorCode, Incident code,)
    - Date_Time: The timestamp for each record.
    - LinkNo: Link number.
    - Volume: vehicles/hour (flow rate)
    - Occupancy: the percentage of time the detector is occupied by vehicles.
    - PlanNo: Real-time timing plan number. In the database, this is linked to a lookup table. PlanNo could be an entity if more attributes, such as phase times, are required to define a timing plan.
    - Color code: the congestion levels determined by the system in Bellevue's TMC.

    Both attributes, Date_Time and LinkNo and are indexed since these two are most often queried.

2. Link(<u>LinkNo,</u> LinkID, LinkLength, SpeedLimit, BeginNode, EndNode, ***CalibratedCoefficients, Predicted Parameters***)

    - LinkNo: Link Number.

- Link ID: stores the info about number of lanes covered by the detector, direction and detector types (system and advance).
- BeginNode and EndNode are the starting and ending intersections, respectively. Each link must be defined by two intersections. These two attributes are foreign keys in the Link table referencing Intersection.IntersectionNo.
- ***CalibratedCoefficients***: This is a set of multiple attributes that stores all the pre-calibrated parameters for roadway link estimation and prediction.
- ***PredictedParameters***: This is a set of multiple attributes that stores all predicted travel times and speeds in different columns.

The details of ***CalibratedCoefficients*** and ***PredictedParameters*** will be explained in the System Design subsection

3. Intersection (<u>IntersectionNo</u>. Longitude, Latitude)
   - Intersection No: Intersection number.
   - Longitude and Latitude identify the location of each intersection.

According to the new design, the database dependency is mitigated by separating the data into different relations (tables).

2.4 System Design

In order to support real-time decision making, the RADAR Net system needs to consider many aspects of an optimized system. The system follows multi-tier architecture design. The technical details of the client-server architecture can be found in *(12)*. As shown in Figure 3, the conceptual system design consists of four layers, offline, online server (middleware), online server (Java Servlet) and online client (Browsers). The tasks are processed in different layers to distribute the computation burden, especially in the server. The function of each layer is explained as follows.

Offline server: the system is designed to estimate and predict traffic parameters. Most algorithms required parameter calibration. The process is mostly done offline using simulations or field observations.

Online server (middleware): This layer is in charge of processing the real-time information commonly used by the online analysis modules. Once the data is downloaded onto the RADAR Net server, loop spot speed estimation and prediction, link speed estimation and link travel time calculations are executed. The calculated data are automatically imported into the database following the designed schemas. In addition to speed data, other traffic parameters, such as predicted volume, can be stored in the database in the same manner. This layer can reduce the computational burden in Java Servlet.

Online server (Java Servlet): The shortest travel-time path algorithm is one of the real-time analysis modules implemented in RADAR Net to support real-time decision. Other RADAR Net statistical analysis modules are also executed here.

Online client layer: This layer handles the requests from all the web browsers visiting the RADAR Net server through World Wide Web (WWW) and visualizes the query results. The code can be executed in the users' browsers using the computing power from each client computer.

Based on this system design, the computational workload can be distributed and lower the server's computational burden. Details of the proposed computational components will be elaborated in the next section.

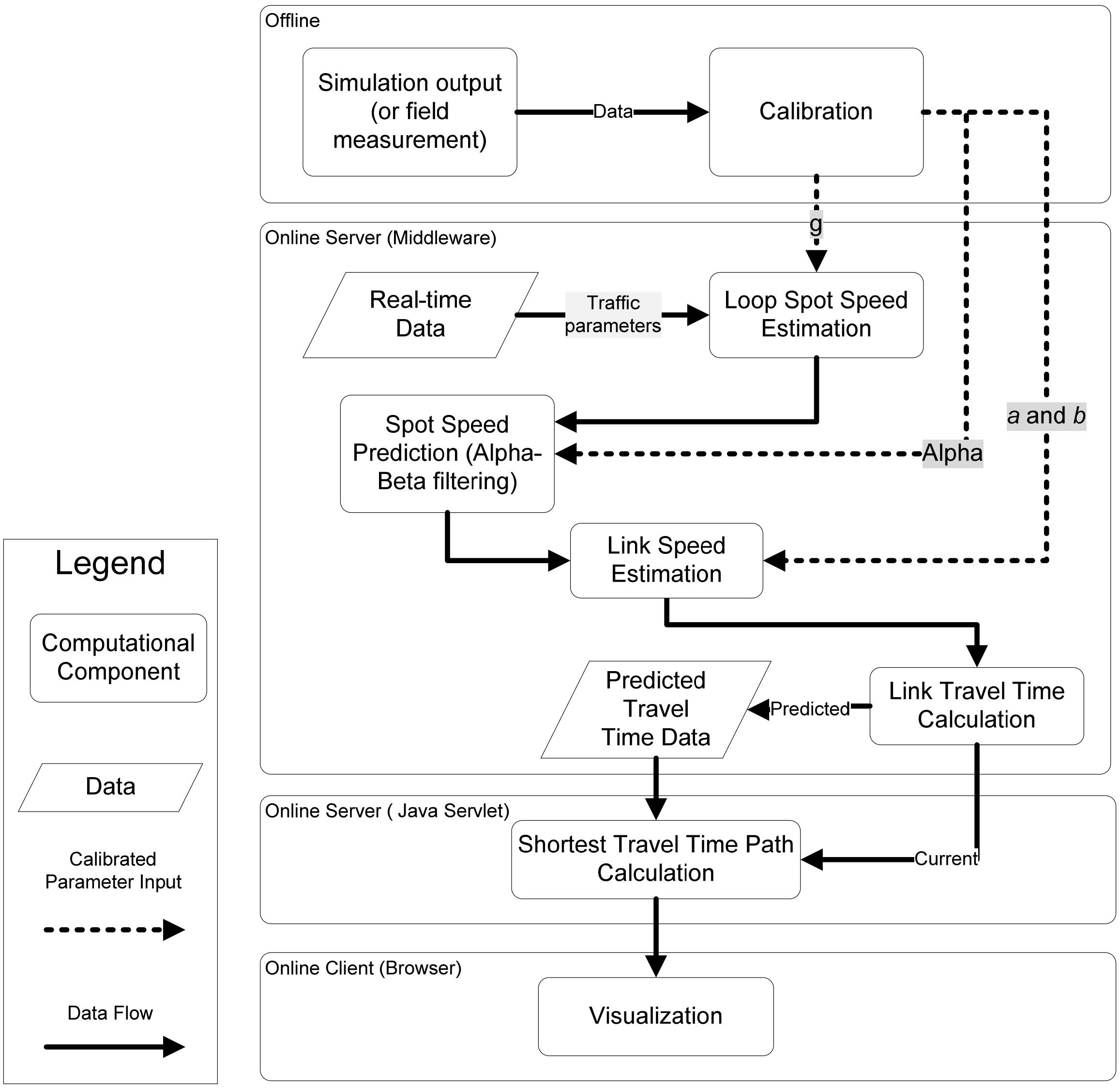


Figure 3 System design

3. SHORT-TERM TRAFFIC PREDICTION

Real-time decision-making relies on an instant, historical and projected overview of the entire arterial network. Hence, the traffic parameters on every link of the arterial network are required to be updated and predicted in a timely manner. Volume, speed and occupancy are considered critical fundamental traffic parameters for a decision making and analysis system. Travel times (link speeds) are regarded as critical information for shortest travel-time path calculations. Shortest travel-time path is a key to emergency vehicle routing that requires the support for a real-time decision making system.

3.1 Loop Spot Speed Estimation

Travel time cannot be measured directly using most existing sensors. Inductance Loop Detectors (ILD) have been commonly used in practice and considered one of the most widely implemented permanent sensors in the U.S. (*13, 14*). The Athol's speed speed estimation formula *(15)*, also called the g-factor approach, is commonly used to estimate the single loop spot speed for freeway (*16*) and arterial (*17*) applications. The loop spot speed for time interval, *t* is defined as

$$S_L(t) = \frac{N(t)}{T \cdot o(t) \cdot g(t)} \qquad (1)$$

with $g(t) = \frac{1}{L(t)}$

where, t is time interval index, N is interval traffic volume, o is occupancy, percentage of time a loop is occupied by vehicles per interval, T is the interval duration; L is the mean effective vehicle length; and g is the speed estimation parameter or called *g*-factor determined by the effective vehicle length. In some application, g is considered time independent (g=2.4) (*18*). For arterials, *g* is considered to be 2.63 (*17*). In our application, g= 2.14 is used assuming the effective vehicle length is affected by transit and trucks (*19*). Equation (1) can be written as

$$S_L(t) = \frac{q(t)}{o(t) \cdot g} \qquad (2)$$

where *q(t)* is the flow rate for time interval *t*.

3.2 Loop Spot Speed Prediction

Many speed prediction models are developed and implemented online, such as the probability-based model by (*20*) and the knowledge-based model by (*21*). However, these models require many inputs, such as signal timing plan. Moreover, a real-time system requires quick response and low computational cost. A dynamic traffic parameter prediction is suitable for our real-time application. Dynamic filtering techniques can not only smooth the real-time data suffering from the random errors but also predict the data in the next state. Among dynamic filtering techniques, the Kalman filter (*22*) has gained attention from system designers because this filter provides high accuracy in prediction and many research projects have demonstrated robustness and reliability for short-term traffic prediction in freeway speeds (*23, 24*), freeway travel time (*25*), arterial travel time (*26*) applications. Recently, Guo et al. (*23*) proposed a Kalman filter-based method to predict freeway speed using single loop detector data. These authors assumed the speed is the state of a discreet time controlled processed governed by the linear stochastic difference equation as follows.

$$S_L(t) = S_L(t-1) + e(t) \quad (3)$$

where $e(t)$ = state process error with mean 0 and variance Q. Next, based on the empirical findings of Guo et al. (*23*), $q(t)/o(t)$, the ratio of flow rate over occupancy (or called *q/o* ratio), has a linear relationship with $S_L(t)$. This linear relationship justifies the application of the Kaman filter. Thus, the measurement equation can be formulated as

$$\frac{q(t)}{o(t)} = gS_L(t) + \varepsilon(t) \quad (4)$$

where $q(t)/o(t)$ = ratio of flow rate over occupancy for time interval t; g = observation parameter (identical to the g-factor in Equation (2)); $\varepsilon(t)$ = observation process error with mean *0* and variance R.

Next, the linear model (Equations (3) and (4)) can be solved by standard Kalman recursion equations and the Kalman gain needs to be calculated recursively based on calibrated g, Q and R(*22*). The method by (*23*) adopted the smoothing function of Kalman filter, neglecting the prediction capability for the state variable because of the purpose of their research. When the data are missing, $S_L(t)$ is supposed to be updated with a predicated value. However, the system state in Equation (3) cannot be updated because this equation lacks a term $u(t-1)$ with a coefficient B to update the speed, $S_L(t)$. (please see (*27*) for more details about Kalman filter). Moreover, variances, R and Q usually need to be calibrated based on real-data and the calibration process will be tedious and cumbersome.

In order to take advantage of the prediction capabilities of Kalman filter and minimize the effort of parameter calibration. The alpha-beta (*α–β*) filter, a simplified version of Kalman filter is used in this study (*28*) for the following reasons:

- The *α–β* filter has been widely applied to object tracking in image processing and can effectively predict the location of the missing objects (*29, 30, 31*),
- Instead of using the positions in the image, the *α–β* filter, is able to mathematically predict speeds mainly based on the measurement of *q/o* (volume/occupancy) ratio. The measurement *q/o* ratio can be regarded as a moving object moving in a one-dimensional line depending on time, *t*,
- The *α–β* filter requires calibration for only one parameter, alpha, and the filter is simplified without computing Kalman gain repetitively.
- Predicting loop spot speeds for the entire arterial network is computationally expensive. The recursive feature of the *α–β* filter can perform in real-time without much burden to the entire system.

In our implementation, every single measurement $x(t)=\frac{q(t)}{o(t)}$ is smoothed and predicted. The α–β filter is defined in the following equations (*28*):

$$x_s(t)=\hat{x}(t\,|\,t)=x_p(t)+\alpha\left[x_o(t)-x_p(t)\right] \quad (5)$$

$$v_s(t)=\hat{\dot{x}}(t\,|\,t)=v_s(t-1)+\frac{\beta}{mT}\left[x_o(t)-x_p(t)\right] \quad (6)$$

$$x_p(t+1)=\hat{x}(t+1\,|\,t)=x_s(t)+T\cdot v_s(t) \quad (7)$$

where $x_o(t)$ = the observed measurement (q/o) at the timestamp *t*; $x_p(t)$ = the predicted measurement at timestamp *t*; $x_s(t)$ = the smoothed measurement at the timestamp *t*; $v_s(t)$ = the smoothed measurement changing rate (It can be regarded as the velocity of the measurement) at the timestamp *t*; T = the sampling interval (T=1 is used since the data is updated every minute); *m* = the number of discrete timestamps since the last measurement; and α, β = fixed-coefficient filter parameters.

The filter starts with an initialization process defined by:

$$x_s(1)=x_p(1)=x_o(1) \text{ and } v_s(1)=0 \quad (8)$$

$$v_s(2)=\frac{x_o(2)-x_o(1)}{T} \quad (9)$$

In order to reduce the calibration effort, the optimal relationship between α and β is known to be (*32*).

$$\beta=2\cdot(2-\alpha)-4\sqrt{1-\alpha} \quad (10)$$

Note that Equations (5)~(9) are used when the measurement can be consistently input into the system. As mentioned in the Introduction section, the data input could be missing due to communication errors or measured speed = 0 (when occupancy or volume=0). In this case, the values of *x* and *v* can be predicted as follows:

$$x_o(t)=x_s(t)=x_p(t) \text{ and } v_s(t)=v_s(t-1) \quad (11)$$

Effectiveness of Prediction

Figure 4 shows the application of the α–β filter on the data collected at the advance loop east of intersection 16 (NE 8th AVE and 106th AVE NE), westbound on NE 8th AVE from 6am to 7pm on July 15th, 2010. Figure 4(a) shows the effectiveness of filter smoothing. Figure 4(b) shows the effectiveness of prediction when 50% of data are missing (randomly removed). The filter still smoothes the predicted measurement while data is missing. However, the area circled in a dotted line shows that the missing multiple data continuously would cause the filter to become increasingly inaccurate. This is also a common prediction constraint for most dynamic filters. In other words, dynamic filters can predict the trend in real-time. If the object is abruptly turning or moving toward other directions, the object tends to be lost. However, the trend can be easily resumed by the filter once the true measurement enters the filter, as illustrated in the prediction results after the circled area.

Calibration

To improve prediction performance, the parameter, α , needs to be selected carefully. According to Equation (5), the higher α is, the more the filter will trust the "correction" from the new measurement. On the other hand, the system will be more sensitive to errors. To demonstrate the feasibility of the α selection process, the calibration process and a sensitivity test for α are conducted in this research. Note that α can be determined based on the characteristics of each link or one single α minimizing the system error can be adopted for the entire network. Either way can be easily implemented offline. This implementation aims to select one single α that can minimize the errors of prediction for the entire network.

Measure of Accuracy

In order to quantify the prediction performance, three measures of accuracy, Mean Absolute Error (MAE), Root Square Mean Error (RSME) and Mean Absolute Percentage Error (MAPE) are used in this study and defined as follows (*33*).

$$MAE = \frac{\sum_{t=1}^{n} |F(t) - G(t)|}{n} \qquad (12)$$

$$RMSE = \sqrt{\frac{\sum_{t=1}^{n} (F(t) - G(t))^2}{n}} \qquad (13)$$

$$MAPE = \frac{1}{n} \sum_{t=1}^{n} \left| \frac{F(t) - G(t)}{G(t)} \right| \qquad (14)$$

where $G(t)$ is the ground truth loop spot speed at time interval t ; $F(t)$ is the predicted link travel at time interval t; n is the total number of samples. In the application, $F(t) = x_s(t)$. If data is missing, $x_o(t) = x_p(t \mid t-1)$ will be smoothed in Equation (5). Even though the prediction error is defined by the difference between $G(t)$ and $F(t)$, the measures of accuracy shows the relative improvement of the smoothing and effectiveness of prediction concurrently since $G(t)$ itself is likely to contain random errors.

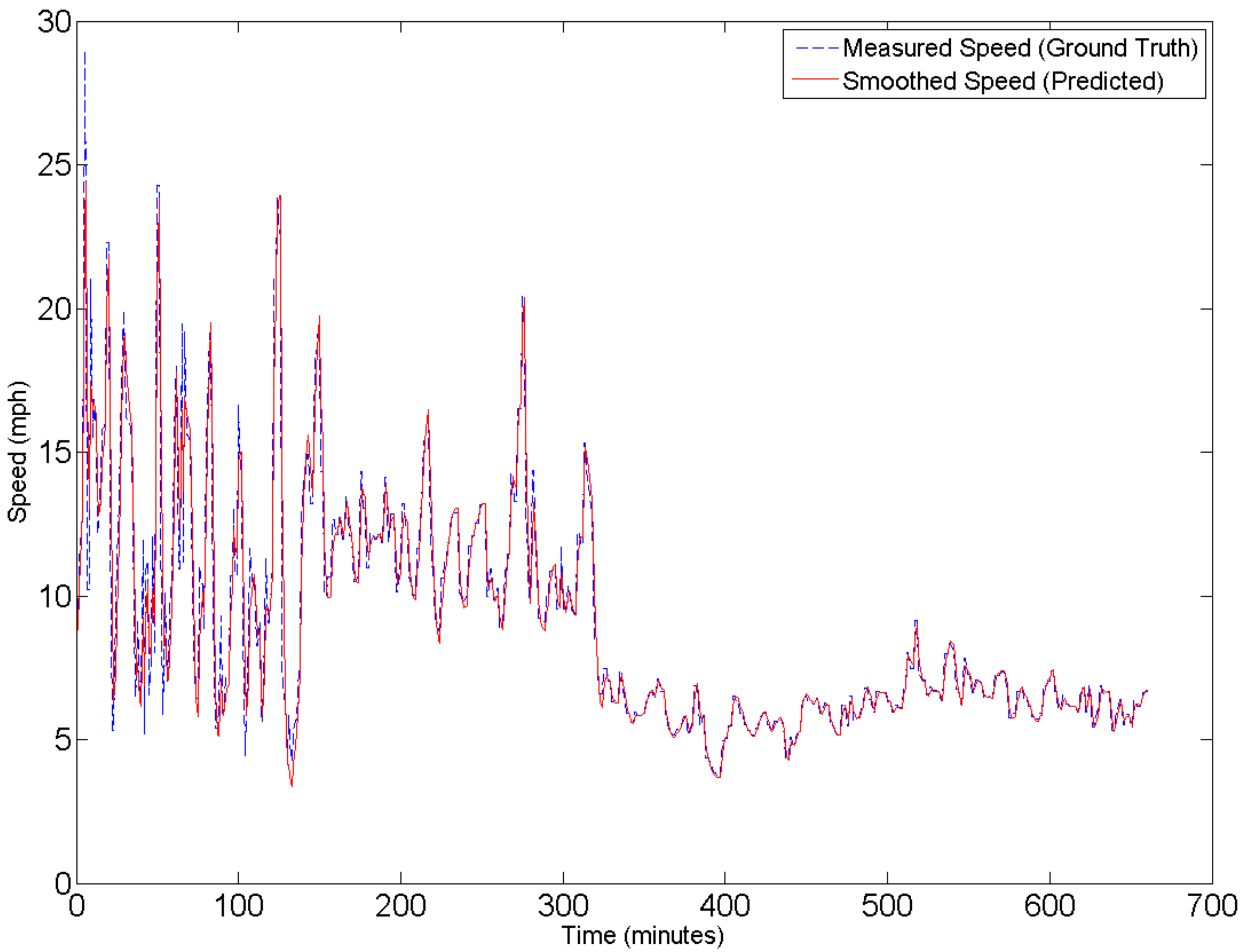
MAE provides an overview of all errors and shows how close the predicted loop speeds are to the ground truth. RMSE shows the average magnitude of the error but penalizes large errors. RMSE indicates the precision of prediction. The MAE and RMSE can be evaluated jointly to determine the variation of the errors. Compared to RMSE, MAPE expresses the error as a percentage without "exaggerating" the error.

Calibration

One day's worth of data collected on July 15 (Thursday), 2010 was extracted from the database. Among 708 links, 472 links with advance detectors on the through movement lanes are adopted in the RADAR Net system. Advance loop data on 23 major arterials links (average volume> 400 veh/h and average occupancy >20%) were selected for the calibration and evaluation process. Data before 6am and after 7pm were excluded in the dataset because few traffic fluctuations are observed during this period and may result in underestimating average prediction errors. In order to determine the most suitable α and effectiveness of prediction, for each data set, different portions of data are randomly removed, ranging from 10% to 90% at 10% increments. The results are sequentially illustrated in Figure 5, from the bottom to the top. As shown in Figure 5(a), if there is no data missing, $\alpha = 0.9$ can result in the "best" results. This is not surprising because the filter "trusts" the new measurement more. It should be worth mentioning that the ground truth measurements may contain random errors. Hence, the ground truth is not "real" ground truth. In this case, the low MAE, RMSE, and MAPE may not absolutely imply that the filter performs better. Therefore, the $\alpha = 0.9$ case in Figures 5(a), 5(b) and 6(c) may imply that the filter is affected by the noise. In contrast, the $\alpha = 0.1$ case shows the "worst" results. It is also reasonable because the filter does not "trust" the new measurement. Figure 5(c) shows a decreasing trend in MAPE, showing the percentage error is reduced when the filter trusts the measurements more. This figure shows there are two drops at $\alpha = 0.2$ and $\alpha = 0.4$, showing these two values could be used if the data quality is poor. Overall, the $\alpha = 0.6$ case shows a drop both in MAE and RSME when the data are 90% missing. This implies $\alpha = 0.6$ is able to reduce the noise and smoothly predict the results concurrently. When 80% of the data is missing, the drop also appears in MAE and RMSE. However, as α increases, the filter becomes more sensitive to noise. Hence, $\alpha = 0.8$ may be used for the links with higher quality of data with low data missing rate. In our application, $\alpha = 0.6$ is selected because this value shows its robustness to consecutive data missing and could minimize the random errors when the measurements are available.

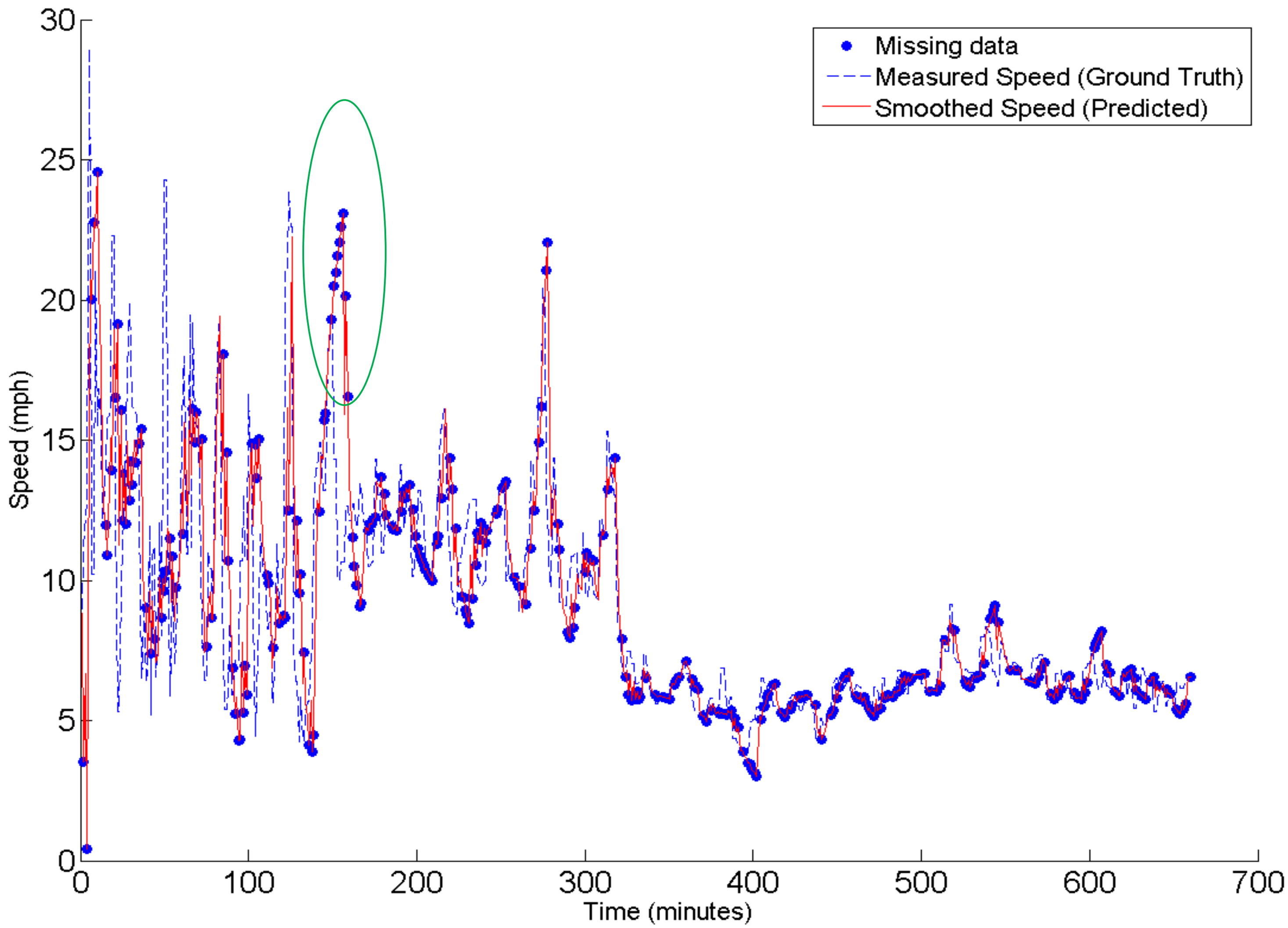
Practical Constraint and Remedy

It is worth noting that the loop-based methods, e.g. (*17*) and (*23*), can perform accurately under congested conditions. Overnight the loop-based method would result in incorrect results due to low volume and occupancy. Therefore, a threshold value has to be determined to separate congested and non-congested conditions. The studies by Guo et al (*23*) and Coifman (*34*) recommended 10% occupancy as an optimal threshold value and this value is used in our system.



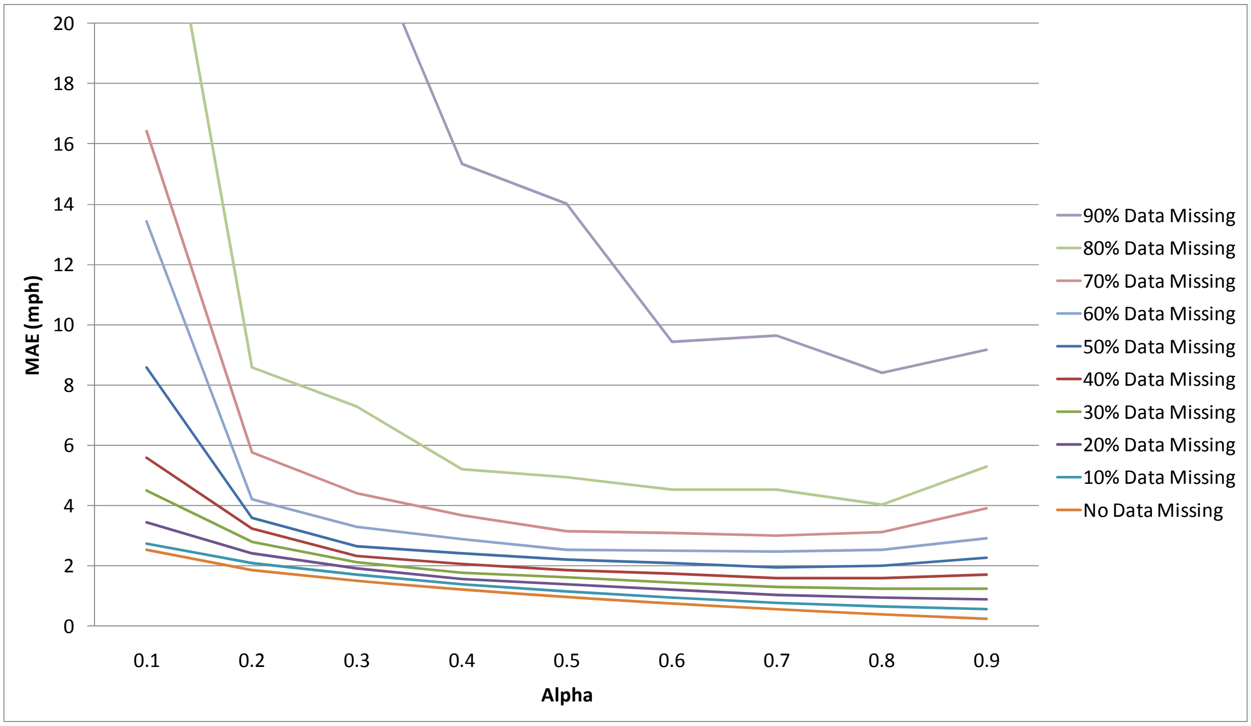
Measured Speed (Ground Truth)
Smoothed Speed (Predicted)
Speed (mph)
Time (minutes)
0
5
10
15
20
25
30
0
100
200
300
400
500
600
700

(a)

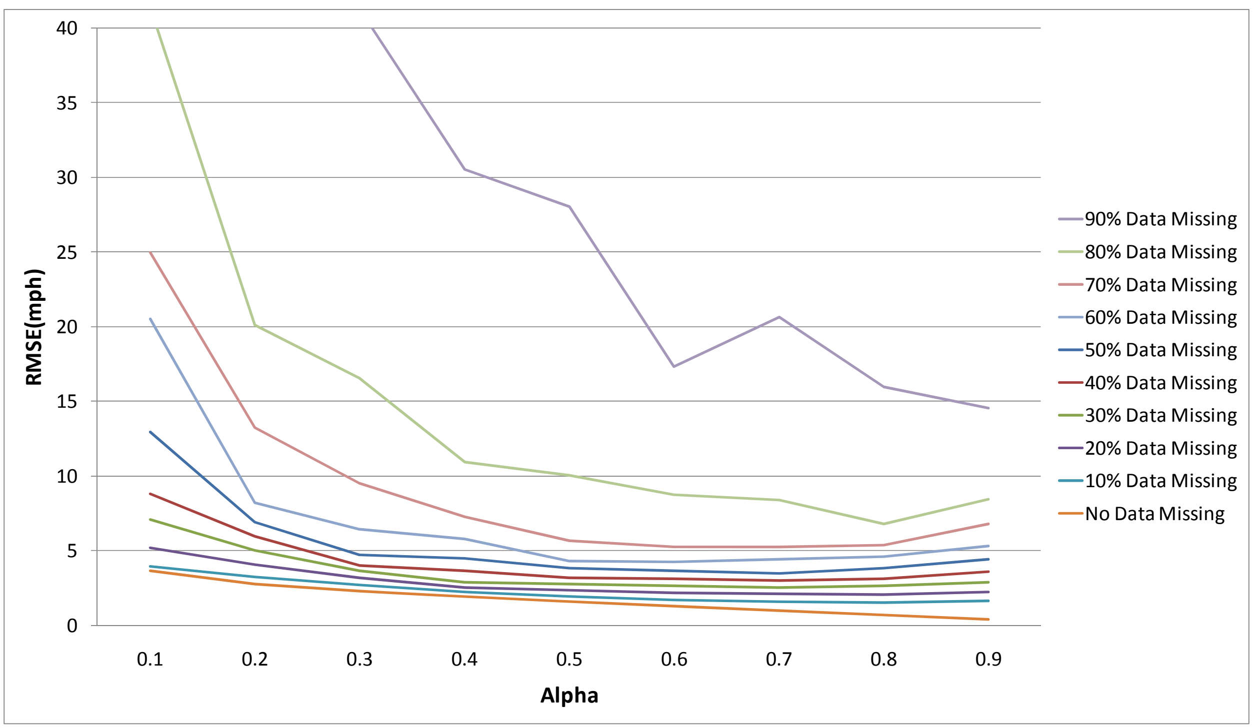


(b)

Figure 4: Application of α–β filter on loop spot speed prediction (α=0.6) (a) No data missing, (b) 50% data missing

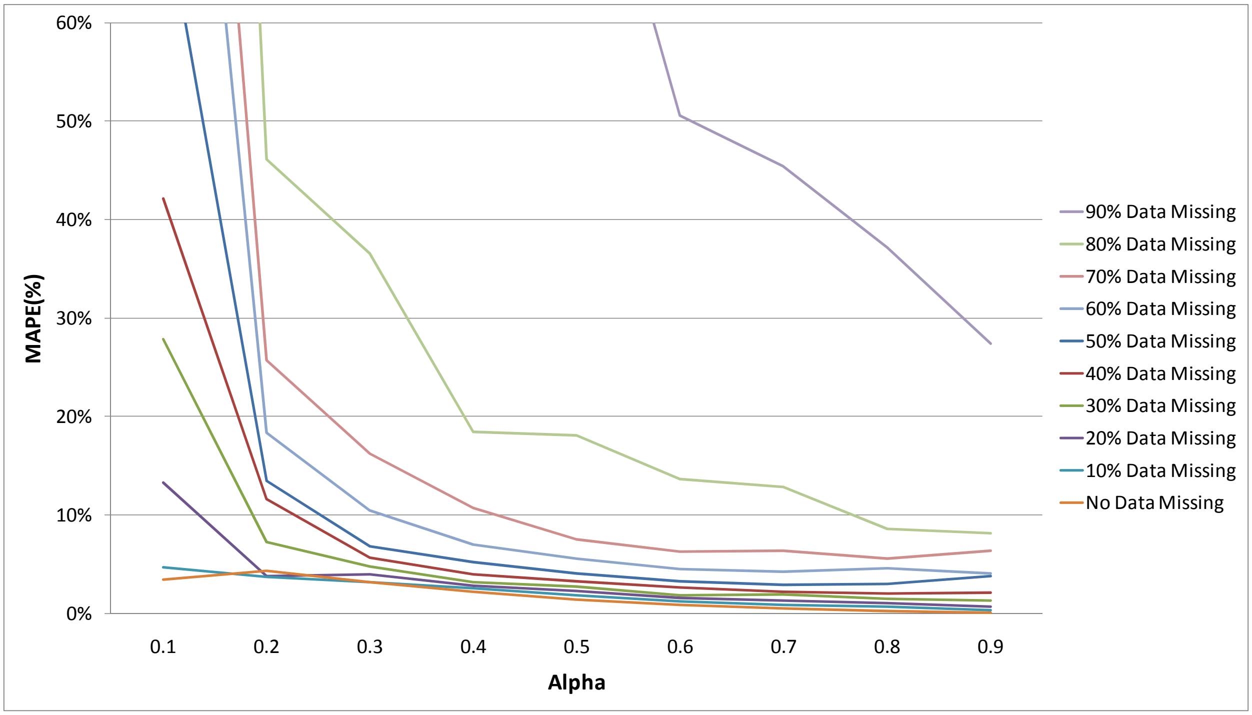


(a)



(b)

Figure 5: Optimal Alpha value selection based on different data missing percentage conditions: (a)The relationship between Alpha and MAE, (b) The relationship between Alpha and RMSE



(c)

Continued, Figure 5: Optimal Alpha value selection based on different data missing percentage conditions: (c)The relationship between Alpha and MAPE under different data missing percentage conditions

3.3 Link Speed Estimation

Once loop spot speeds are estimated and predicted, loop spot speeds can be converted to link speeds to better represent the link performance. Based on Zhang's research, loop spot speed can be representative of link speed in congested conditions. It was found in (*19*) that the advance loop spot speed is likely to overestimate the link speed if the ground truth link speed is higher. Hence, the model to represent the relationship between the ground truth link speed and advance loop spot speed is formulated as (*19*):

$$\hat{S}_J(t) = a\hat{S}_L(t) - \exp(b\hat{S}_L(t)) + 1 \qquad (15)$$

where $\hat{S}_J(t)$ is the estimated link speed, $\hat{S}_L(t)$ is the loop spot speed at the advance detector, and a and b are coefficients that require calibration. The constant value, 1, allows the calibrated model to traverse the origin (0,0). In other words, when $\hat{S}_L(t) = 0$, $\hat{S}_J(t) = 0$. This is based on the assumption that the spot speed should be equal to the link speed when the measured spot speed is close to zero. (*17*). The parameters, a and b are calibrated by the traffic simulation software package, VISSIM model (*35*). As implemented, the universal parameters a and b (1.0 and 0.05, respectively) are calibrated based on the major streets and applied to all links to demonstrate the feasibility of the approach. To achieve most accurate results, the calibration should be conducted for every link. After link speeds are available, the link travel time can be easily calculated using known link lengths. Since the link travel speed estimation takes signal control into account (*19*), the calculated link travel time also contains the control delay. Therefore, the route travel time is simply the summation of all links along the route.

4. SYSTEM IMPLEMENTATION

4.1 Implementation

Based on the system design shown in Figure 3, the online client and online server (Java Servlet) layers of the RADAR Net system is programmed using Google Web-Toolkit (GWT) (*36*) combined with Eclipse (*37*), an open-source Java IDE. Compared with the previous GATI system development environment, the development efficiency has been greatly improved. The code is optimized and converted to the JavaScript code by GWT. The online server layer is implemented in C#. The server runs on Windows Server 2008 operating system (OS) with MS SQL Server 2008.

4.2 Application Modules

Five application modules have been implemented in the RADAR system to facilitate real-time decision making: 1) Arterial real-time map, 2) Arterial data analysis, 3) Historical arterial map query, 4) Dynamic shortest travel time routing, and 5) Arterial data sharing. Modules (1)~(3) were the re-implementation based on the GATI system (*9*) with improved database design and performance. Figure 6(a) shows the volume-occupancy scatter plot during June, 20th ~ June 26th, 2010. The statistical analysis are calculated online once the "statistical analysis button is clicked. Figure 6(b) shows the time domain plot of the volume and occupancy data during the same period. The scalable visualization bar can easily zoom into June, 23th and slide the window to investigate traffic variation. Figure 6(c) shows the real-time traffic map. One can notice that the links between Intersection A and B are not all experiencing free-flow conditions. Figure 6(d) shows the shortest path between Intersections A and B. The shortest (travel time) path is calculated by the A* algorithm (*38*) based on real-time link speeds updated by the α–β filter. Since the speed data are stored in the database, the algorithm can be executed in real-time once two nodes are selected. The path successfully skips the congested links indicated in Figure 6(c). All these modules can be used to investigate various key issues. For example, the shortest route module can be combined with the arterial data analysis module to investigate the causes of the bottleneck.

4.3 Performance

The database design effectively reduces the query time for loop spot speed estimation and meanwhile increases the online algorithm performance. The query for retrieving all the attributes of the entire network by joining all tables takes less than 500ms. Downloading the raw data, calculating the link speeds, and updating the travel time data for all links in the network takes less than an average of two seconds in the middleware. Moreover, the shortest path algorithm executed within the CBD area can be calculated in an average of one second.

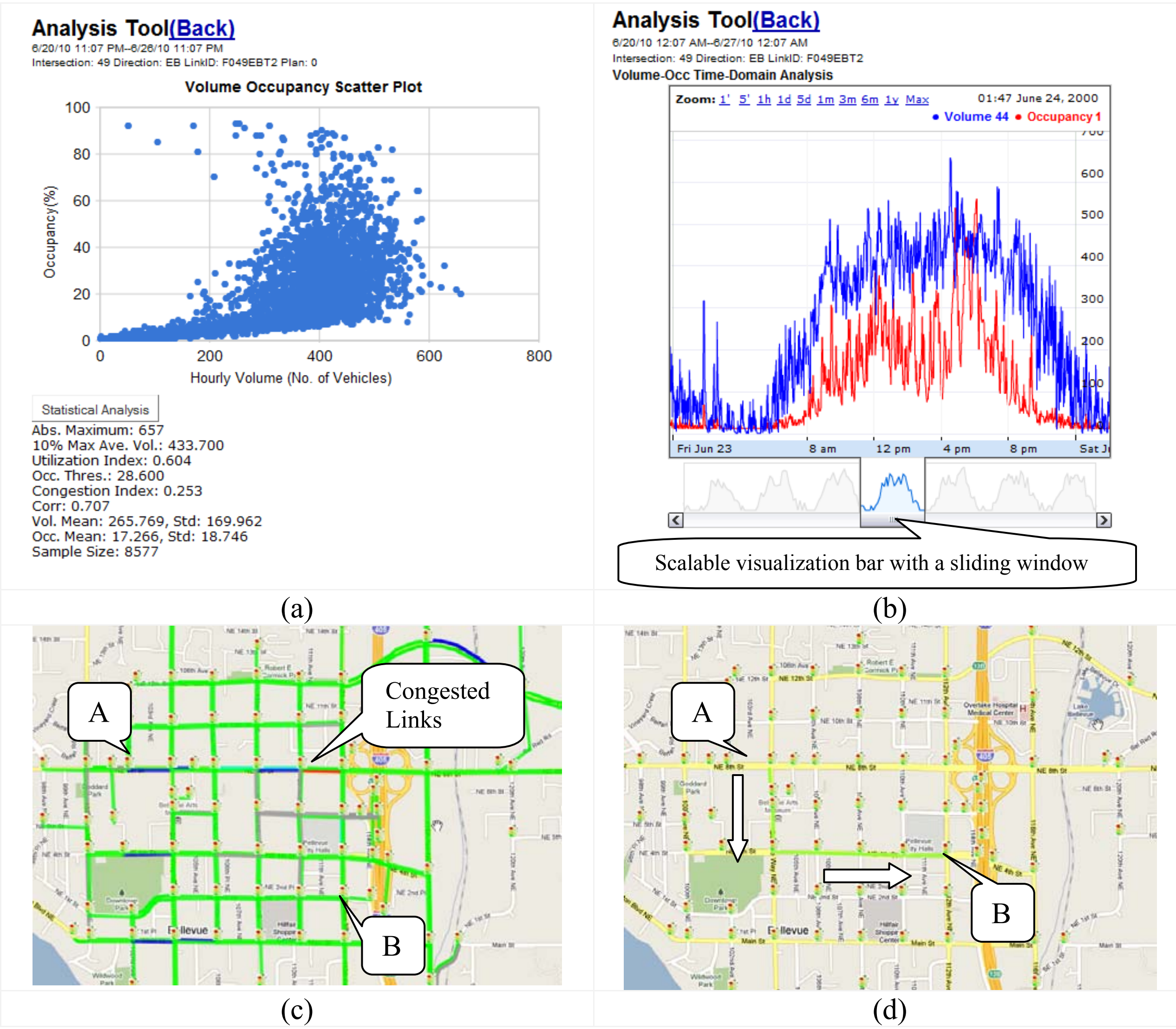


Figure 6 RADAR Net modules (a) Volume and occupancy scatter plot and analysis (June, 20th ~ June 26th, 2010), (b) Scalable time-domain plot ((June, 20th ~ June 26th, 2010), (c)Real-time traffic flow map at 5:45pm, July 29th 2010 (Thursday), and (d) real-time dynamic shortest travel time routing

5. CONCLUSIONS AND RECOMMENDATIONS

System performance and simple implementation are the keys to the success of a real-time decision making system for large urban arterial network. Many practical challenges hurdle the development of such a system. This paper presents a web-based Real-time Analysis and Decision-making for ARterial Network) RADAR Net system that demonstrates the computational capability between server and clients. A practical and scalable arterial database design is also proposed. The schemas can be used as a template for storing arterial data for other agencies. Since the database design is based on the relational model, this design can incorporate more other arterial data and increase the system scalability. The RADAR Net system contains four layers: offline server, online server (middleware) and online server (Java Servlet) and online client. This four-layer system design successfully distributes the computational burden of the system. Traffic parameters are calculated or retrieved directly from the loop detector. The RADAR system can dynamically predict and smooth real-time loop spot speeds by using α–β filter, a simplified version of Kalman filter while maintaining high system performance. Many application modules are implemented based on the current system architecture and prove feasible to perform real-time analysis and assist decision making.

For an urban arterial network, travel time (speed) is an important indicator for the traffic state. Currently, only the arterial traffic parameters (e.g. volume and occupancy) are the main inputs to the system for travel time estimation. It is also recommended that more datasets (e.g. incident data) should be included in the future analysis module development in order to provide more comprehensive decision support. With more data sets, the RADAR Net system can accomplish additional potential real-time applications, e.g. emergency evacuation and emergency vehicle routing and bottleneck analysis. These additional modules can be easily implemented based on the existing development. In addition to computational performance evolution, the system should be further evaluated by using several case studies, e.g. how the recurrent and non-recurrent congestion affect theoretical shortest paths (route choice). A user review process is another feasible option to evaluate how effectively the system supports decision making.

Even though the RADAR Net system demonstrates its capability of real-time data processing, the system still has some limitations. For example, the prediction function cannot deal with long term missing data and malfunction in loop detectors. For those cities without loop detector infrastructure or with missing detectors in some roadway links, the advanced sensor technologies, such as Bluetooth travel time detectors can be used to provide missing travel time data. Moreover, the performance may be reduced if thousands of queries are executed simultaneously. One possible solution is to use concurrency control. In addition, the database design can be further improved by incorporating multi-dimensional databases into the system to handle aggregated data in real-time. To increase the computing power of RADAR Net, cloud computing could be a potential solution.

ACKNOWLEGMENTS

The authors would like to thank Fred Liang, Mike Whiteaker and Brook Durant in the City of Bellevue for providing arterial traffic data. Special thanks to Mengqi Wu for coding the first generation of RADAR Net and thanks Aaron Knight for updating the network information in the database. The authors are grateful to Jonathan Corey for database maintenance and support.

REFERENCES

1. Hu, P., B. Boundy, T. Truett, E. Chang, and S. Gordon. *Cross-cutting Studies and State-of-the-practice Reviews : Archive and Use of ITS-generated Data*. Publication Oak Ridge National Laboratory. Center for Transportation Analysis. 2002.
2. Seymour, E.J., and B. Miller. Use of A Google-Based Mapping System for Traffic Management Center Web Navigation. In Transportation Research Board 86th Annual Meeting. CD-ROM. Washington DC, United States, 2007.

3. Chen, C. *Freeway Performance Measurement System (PeMS)*. Publication UCB-ITS-PRR-2003-22. UC Berkeley: California Partners for Advanced Transit and Highways (PATH). 2003.
4. Bertini, R., S. Hansen, A. Byrd, and T. Yin. Experience Implementing a User Service for Archived Intelligent Transportation Systems Data. *Transportation Research Record: Journal of the Transportation Research Board*, Vol. 1917, 2005, pp. 90-99.
5. Pack, M.L., J.R. Bryan, and A. Steffes. *Overview and Status of Regional Integrated Transportation Information System in the National Capital Region*. In Transportation Research Board 87th Annual Meeting. CD-ROM. 2008.
6. Petty, K., J. Kwon, and A. Skabardonis. Arterial Performance Measurement System—APeMS. In Workshop on Performance Measures, 85th TRB Annual Meeting. CD-ROM. Washington D.C., 2005.
7. Cheevarunothai, P., Y. Wang, and N. Nihan. Identification and Correction of Dual-Loop Sensitivity Problems. *Transportation Research Record: Journal of the Transportation Research Board*, Vol. 1945, 2006, pp. 73-81.
8. Wu, Y.-J., Y. Wang, and D. Qian, A Google-Map-Based Arterial Traffic Information System, *IEEE Intelligent Transportation Systems Conference*, 2007, pp. 968-973.
9. Wu, Y.-J., and Y. Wang. An Interactive Web-based System for Urban Traffic Data Analysis. *International Journal of Web Applications*, Vol. 1, No. 4, 2009, pp. 241-252.
10. Codd, E.F. A relational model of data for large shared data banks. *Commun. ACM*, Vol. 13, No. 6, 1970, pp. 377-387.
11. Garcia-Molina, H., J.D. Ullman, and J. Widom, *Database systems: the complete book*, Pearson Prentice Hall, Upper Saddle River, N.J., 2009.
12. Ma, X., Y.-J. Wu, and Y. Wang. An E-Science of Transportation Platform for Data Sharing, Visualization, Modeling, and Analysis. In Transportation Research Board 90th Annual Meeting. CD-ROM. 2011.
13. Klein, L.A., *Sensor Technologies and Data Requirements for ITS*, Artech House, Boston, 2001.
14. Klein, L.A., M.K. Mills, and D.R.P. Gibson, *Traffic Detector Handbook*, U.S. Dept. of Transportation, Federal Highway Administration, Research, Development, and Technology, Turner-Fairbank Highway Research Center, McLean, VA, 2006.
15. Athol, P. Interdependence of Certain Operational Characteristics Within a Moving Traffic Stream. *Highway Research Record 72*, 1965, pp. 58–87.
16. Wang, Y., and N.L. Nihan. Can Single-Loop Detectors Do the Work of Dual-Loop Detectors? *Journal of Transportation Engineering*, Vol. 129, No. 2, 2003, pp. 169-176.
17. Zhang, H.M. Link-Journey-Speed Model for Arterial Traffic. *Transportation Research Record: Journal of Transportation Research Board*, No. 1676, 1999, pp. 109-115.
18. Ishimaru, J.M., and M.E. Hallenbeck. *Flow Evaluation Design Technical Report*. Publication WA-RD 466.2. W.D.o. Transportation, Washington Department of Transportation. 1999.
19. Wu, Y.-J., G. Zhang, and Y. Wang, Link Journey Speed Estimation for Urban Arterial Performance Measurement using Advance Loop Detector Data under Congested Conditions, In Transportation Research Board 90th Annual Meeting. CD-ROM. 2011.
20. Lin, W.-H., A. Kulkarni, and P. Mirchandani. Short-Term Arterial Travel Time Prediction for Advanced Traveler Information Systems. *Journal of Intelligent Transportation Systems: Technology, Planning, and Operations*, Vol. 8, No. 3, 2004, pp. 143 - 154.

21. Lee, W.-H., S.-S. Tseng, and S.-H. Tsai. A knowledge based real-time travel time prediction system for urban network. *Expert Systems with Applications*, Vol. 36, No. 3, Part 1, 2009, pp. 4239-4247.
22. Kalman, R.E. A New Approach to Linear Filtering and Prediction Problems. *Transactions of the ASME: Journal of Basic Engineering*, Vol. 82, 1960, pp. 35-45.
23. Guo, J., J. Xia, and B.L. Smith. Kalman Filter Approach to Speed Estimation Using Single Loop Detector Measurements under Congested Conditions. *Journal of Transportation Engineering*, Vol. 135, No. 12, 2009, pp. 927-934.
24. Yang, F., Z. Yin, H. Liu, and B. Ran. Online Recursive Algorithm for Short-Term Traffic Prediction. *Transportation Research Record: Journal of the Transportation Research Board*, Vol. 1879, 2004, pp. 1-8.
25. Zhang, X., and J.A. Rice. Short-term Travel Time Prediction. *Transportation Research Part C: Emerging Technologies*, Vol. 11, No. 3-4, 2003, pp. 187-210.
26. Liu, Y., P.-W. Lin, X. Lai, G.-L. Chang, and A. Marquess. Developments and Applications of Simulation-Based Online Travel Time Prediction System: Traveling to Ocean City, Maryland. *Transportation Research Record: Journal of the Transportation Research Board*, Vol. 1959, 2006, pp. 92-104.
27. Welch, G., and G. Bishop. *An Introduction to the Kalman Filter*. http://www.ing.unife.it/docenti/SilvioSimani/www/KF_lesson.html. Accessed April 1st, 2010.
28. Blackman, S.S., *Multiple Target Tracking with Radar Applications*, Artec House, Inc., Norwood, MA, 1986.
29. Malinovskiy, Y., Y.-J. Wu, and Y. Wang. Video-Based Monitoring of Pedestrian Movements at Signalized Intersections. *Transportation Research Record: Journal of the Transportation Research Board*, Vol. 2073, 2008, pp. 11-17.
30. Wu, Y.-J., F.-L. Lian, and T.-H. Chang, Traffic Monitoring and Vehicle Tracking using Roadside Cameras, *Systems, Man and Cybernetics, 2006. SMC '06. IEEE International Conference on*, 2006, pp. 4631-4636.
31. Wu, Y.-J., F.-L. Lian, C.-P. Huang, and T.-H. Chang. Image Processing Techniques for Lane-Related Information Extraction and Multi-Vehicle Detection in Intelligent Highway Vehicles. *International journal of Automotive Technology* Vol. 8, No. 4, 2007.
32. Kalata, P.R. The Tracking Index: A Generalized Parameter for α-β and α-β-γ Target Trackers. *Aerospace and Electronic Systems, IEEE Transactions on*, Vol. AES-20, No. 2, 1984, pp. 174-182.
33. Washington, S., M.G. Karlaftis, and F.L. Mannering, Statistical and Econometric Methods for Transportation Data Analysis, Chapman & Hall/CRC, Boca Raton, 2003.
34. Coifman, B. Improved velocity estimation using single loop detectors. *Transportation Research Part A: Policy and Practice*, Vol. 35, No. 10, 2001, pp. 863-880.
35. PTV, *Vissim User Manual: Version 5.2*, Karlsruhe, Germany, 2010.
36. Google Inc. *Google Web Toolkit*. http://code.google.com/webtoolkit/. Accessed July, 29 2010.
37. The Eclipse Foundation. *Eclipse*. http://www.eclipse.org/. Accessed July 29 2010.
38. Hart, P.E., N.J. Nilsson, and B. Raphael. A Formal Basis for the Heuristic Determination of Minimum Cost Paths. *Systems Science and Cybernetics, IEEE Transactions on*, Vol. 4, No. 2, 1968, pp. 100-107.