Development of Web-Based Analysis System for Real-Time Decision Support on Arterial Networks

Yao-Jan Wu, Shi An, Xiaolei Ma, and Yinhai Wang

With increasing amounts of data being collected for intelligent transportation systems on arterial networks, the archival, management, and analysis of complex network traffic data have become a challenge. Challenges include inconsistent data connections, data quality control, query performance, traffic prediction, and computational limitations. The web-based RADAR Net system is presented to address these challenges. This system adopts a relational database with link, intersection, and detector entities. Relational data demonstrate 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 on the server. To monitor arterial performance, link speeds are calculated directly from loop detector data retrieved in Bellevue, Washington. The system can dynamically predict and smooth real-time loop spot speeds by using an alpha–beta filter (a simplified version of the Kalman filter) while maintaining high system performance. The link speeds of the entire network are calculated and updated in real time. Many application modules (e.g., for capacity analysis and dynamic routing) based on the system architecture have been implemented and have proven the system feasible for performing real-time analysis and assisting decision making.

With new technology developments in intelligent transportation systems, 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 systems because it requires additional effort to clean, archive, analyze, and interpret the data that describe complex traffic conditions. Raw data gathered from sophisticated sensor networks require further processing to produce useful results for traffic management and traveler information systems.

Y.-J. Wu, Department of Civil and Environmental Engineering, University of Virginia, Thornton Hall B228, 351 McCormick Road, Charlottesville, VA 22904-1000. S. An, School of Transportation Science and Engineering, Harbin Institute of Technology, 202 Haihe Road, Nangang District, Harbin 150090, China. X. Ma and Y. Wang, Department of Civil and Environmental Engineering, University of Washington, Seattle, WA 98195-2700. Corresponding author: Y.-J. Wu, yaojan@virginia.edu.

Transportation Research Record: Journal of the Transportation Research Board, No. 2215, Transportation Research Board of the National Academies, Washington, D.C., 2011, pp. 24–36.
DOI: 10.3141/2215-03

Managing and using traffic sensor data effectively has been a challenging issue. For example, traffic sensor data must be processed before being transferred to advanced traveler information systems (ATISs). However, an ATIS cannot be informative and successful without a well-designed database and data-analysis methods. To support real-time information display and historical data analysis, the idea of an Archived Data User Service (ADUS) has been proposed since the 1990s, allowing transportation agencies to efficiently store and redistribute data by intelligent transportation systems generated for analysis (*1*).

With improvements in information technology, web-based systems have become popular (e.g., *2–4*) because they can efficiently display, analyze, and disseminate traffic information in a timely manner. Because of these advantages, a web-based system is suitable for supporting real-time decision making, such as for developing and executing emergency evacuation plans and routing emergency vehicles. However, currently, most web-based ATISs and ADUSs focus on freeway applications, and few address issues on urban streets. For example, Chen (*3*) and Bertini et al. (*4*) developed two major ATISs with ADUSs for freeway applications: the Freeway Performance Measurement System (PeMS) and Portland Regional Transportation Archive Listing, respectively. Pack et al. developed the Regional Integrated Transportation Information System—one of the most comprehensive ATISs and ADUSs that demonstrates capabilities of regionwide automated data sharing, dissemination, and archival (*5*). However, their effort for real-time arterial data processing was not addressed. Petty et al. recently developed the arterial performance measurement system (APeMS), but system functionality was limited and incapable of analyzing a large-scale arterial network (*6*).

On the basis of current state of the art, most system development has difficulty providing real-time, networkwide functions for arterial analysis and decision support, possibly because of the lack of real-time arterial data. Hence, researchers have not fully explored techniques for real-time arterial data processing and analysis. Nowadays, more and more cities (e.g., Bellevue, Washington) are capable of providing high-resolution arterial traffic data. If data can be processed in a timely manner, the real-time results can assist road users and engineers in real-time decision making in a complicated arterial network while providing 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. The prevailing challenges for such a system include the following:

1. Inconsistent data connections. There are several ways to transmit real-time data between the data providers and clients. For example, the Washington State Department of Transportation adopts Simple Object Access Protocol to disseminate real-time extensible markup language data for incidents. The City of Bellevue, Washington, archives traffic data as flat files in the data server, and the public can fetch data via file transfer protocol. Regardless of data-transmission methods, data could be missing while being transmitted from the on-site sensors to the traffic management center. In practice, communication fails periodically.

2. Data quality control. A procedure for the quality control of data is essential to provide accurate results. Some erroneous data should be removed. For example, loop detectors generally have sensitivity errors that result in incorrect detection readings (*7*). Moreover, speed estimation data should be corrected in situations where occupancy or flow rate 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 an improved infrastructure for collection of intelligent transportation system data huge amounts of data are transmitted to the data warehouse. An efficient database design is key to improving query performance.

4. Traffic prediction. Traffic status changes dynamically with some randomness. Short-term prediction has been a critical issue. Most prediction algorithms require high computational power and are unsuitable for implementation in a real-time system. Most decision-making processes (e.g., shortest-path routing estimation) require smoothing and prediction processes for the detector measurements (e.g., flow rate or speed). 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 must be tolerant of noise and error to minimize the impact of erroneous data.

5. Computational limitation. Calculating statistics and implementing algorithms requires computational power. If the computation burden is only on the server side, server performance will be affected. 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.

In an effort 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 the web-based RADAR Net system. The remainder of the paper is organized as follows. First, the database and system designs of RADAR Net are introduced. Next, details of the design flow are described, followed by a detailed consideration of the estimation and prediction of loop spot speed and the calculation of link speed. The implementation of each functional module and system performance are discussed. The paper concludes with lessons learned, recommendations, and future work.

RADAR Net is implemented as a subsystem of the Digital Roadway and Interactive Visualization and Evaluation Network (DRIVE Net; http://www.uwdrive.net), an online interdisciplinary data integration and analysis platform hosted by the Smart Transportation and Application Research Laboratory at the University of Washington.

FRAMEWORK AND SYSTEM DESIGN

Network Description

As of July 2010, the City of Bellevue was operating 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. Real-time traffic data (e.g., flow rate and occupancy) are mainly retrieved from advance loop detectors located 100 to 130 ft (30.5 to 45.7 m) upstream from the stop bar of each approach. All data are stored in an file transfer protocol data server and downloaded automatically to the DRIVE Net arterial database every minute. More details of the data-retrieval process can be found elsewhere (*8, 9*). The intersections with real-time data are displayed in the traffic light icons in Figure 1. As of July 15, 2010, real-time traffic data from 706 loop detectors were being sent back to Bellevue's traffic management center.

Previous Work

The Google Maps–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, and scatter and time–domain plots for flow rate and occupancy. The analytical statistics can be calculated online depending on user input. However, this system suffers from several drawbacks:

- The GATI system is programmed in JavaScript and PHP. Few integrated development environment software packages are designed for JavaScript and PHP. The debugging process tends to be slow and tedious.
- Few codes and libraries can be found and reused, even though JavaScript and PHP are object-oriented. 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. The interface is difficult to adjust and fit to all types of browser settings.
- The visualization module was completed with third-party packages, and visualization flexibility is limited.
- The database has a dependency issue that increases the database size.

These issues also are commonly observed in practical applications. To address them, RADAR Net aims to renovate the GATI system by improving system and database designs.

Database Design

Real-time decision making relies on prompt query response from databases. For a typical online transaction processing system, database design is key to retrieving timely data through query. RADAR Net uses the relational database, which is commonly used for such systems and can provide many advantages (*10*). For example, data are organized in different relations (tables) and use structured query language to query the specific results as desired (*11*). Moreover, new relations and attributes can be added to the design easily to increase design flexibility and database scalability.

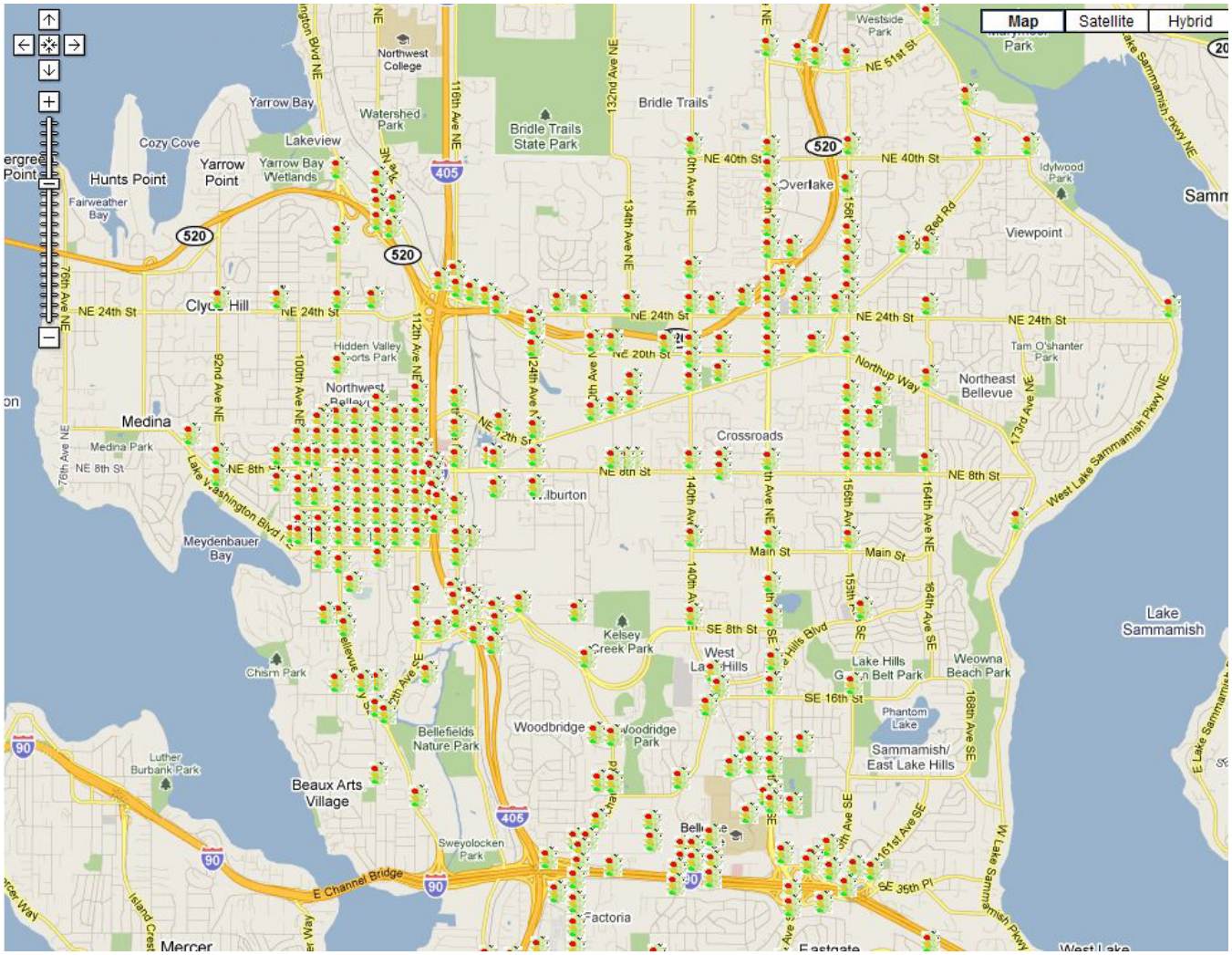


FIGURE 1 Arterial network studied in Bellevue. (Source: Background image is from http://maps.google.com).

The database design proposed in previous research was found to contain data dependency and anomalies (*8*). Redundant data occupied more than 40% of the storage space. Therefore, the entity–relationship (E-R) diagram was improved as shown in Figure 2.

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 structural query language database. The detector table stores the real-time detector data. The link and intersection tables store the time-independent attributes. Thus, users can add or update links or intersections without affecting the detector table. The attributes are briefly explained in the following lists (underscored attributes indicate keys in the database design).

1. Detector (<u>Date_Time</u>, <u>LinkNo</u>, FlowRate, Occupancy, PlanNo, Cycle Length, ColorCode, Incident code):
   – Date_Time is the time stamp for each record;
   – LinkNo is the link number;
   – FlowRate is the number of vehicles per hour (flow rate);
   – Occupancy is the percentage of time the detector is occupied by vehicles;
   – PlanNo is the real-time timing plan number (linked to a lookup table in the database), which could be an entity if more attributes (e.g., phase times) are required to define a timing plan; and
   – Color_code is the congestion level determined by the system in Bellevue's TMC.

Both attributes Date_Time and LinkNo are indexed because they are most often queried.

2. Link(<u>LinkNo</u>, LinkID, LinkLength, SpeedLimit, BeginNode, EndNode, <u>CalibratedCoefficients</u>, <u>PredictedParameters</u>):
   – LinkNo is the link number;
   – LinkID 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 (because each link must be defined by two intersections), respectively, and foreign keys in the Link table referencing IntersectionNo;
   – CalibratedCoefficients is a set of multiple attributes that stores all the precalibrated parameters for roadway link estimation and prediction; and
   – PredictedParameters is a set of multiple attributes that stores all predicted travel times and speeds in different columns.

The details of CalibratedCoefficients and PredictedParameters are explained in the subsection on system design.

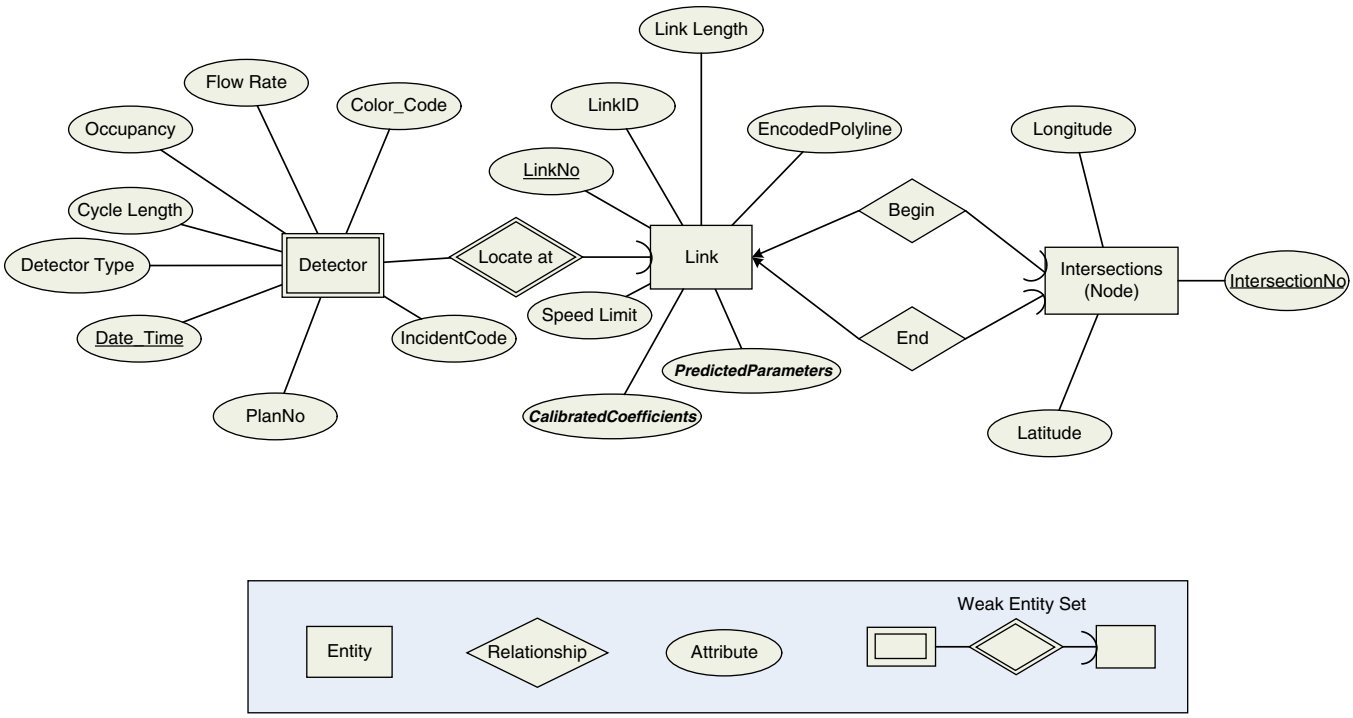


FIGURE 2 E-R diagram design for the arterial network database.

3. Intersection (IntersectionNo, Longitude, Latitude):
   – IntersectionNo is the intersection number and
   – Longitude and Latitude denote the location of each intersection.

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

System Design

In support of real-time decision making, RADAR Net must consider many aspects of an optimized system. The system consists of a multi-tier design; technical details of the client–server architecture can be found elsewhere (*12*). As shown in Figure 3, the conceptual system design consists of four layers: offline server, online server (middleware), online server (Java Servlet) and online client (browser). Tasks are processed in different layers to distribute the computation burden, especially on the server. The layer functions are explained as follows:

- The offline server is designed to estimate and predict traffic parameters. Most algorithms require parameter calibration. The process is done mostly offline, using simulations or field observations.
- The online server (middleware) processes the real-time information commonly used by the online analysis modules. After data are downloaded to the RADAR Net server, loop spot speed is estimated and predicted, link speed is estimated, and link travel time is calculated. Calculated data are automatically imported into the database according to the designed schemas. In addition to speed data, other traffic parameters (e.g., predicted flow rate) can be stored in the database in the same manner. This layer can reduce the computational burden in Java Servlet.
- The online server (Java Servlet) layer facilitates real-time response to client requests. For example, the algorithm for the shortest (travel time) path is one of the real-time analysis modules implemented in this layer to support real-time decision making in RADAR Net. Other RADAR Net statistical analysis modules also are executed in this layer.
- The online client layer handles the requests from all the web browsers visiting the RADAR Net server through the World Wide Web and visualizes the query results. The code can be executed in the user's browser using the computing power of the client's computer.

With this system design, the computational workload is distributed, thereby lowering the computational burden on the server. Details of the proposed computational components are elaborated in the next section.

PREDICTION OF SHORT-TERM TRAFFIC

Real-time decision making relies on an instant, historical, and projected overview of the entire arterial network. Hence, traffic parameters on every link of the arterial network must be updated and predicted in a timely manner. Flow rate, speed, and occupancy are considered critical fundamental traffic parameters in a system used for decision making and analysis. Travel time (link speed) information is critical for calculating the shortest path, which is essential to emergency vehicle routing and requires support for a real-time decision-making system.

Estimation of Loop Spot Speed

Travel time cannot be measured directly using most existing sensors. Inductance loop detectors commonly have been used in practice and are considered one of the most widely implemented types of permanent sensors in the United States (*13, 14*). Athol's speed estimation

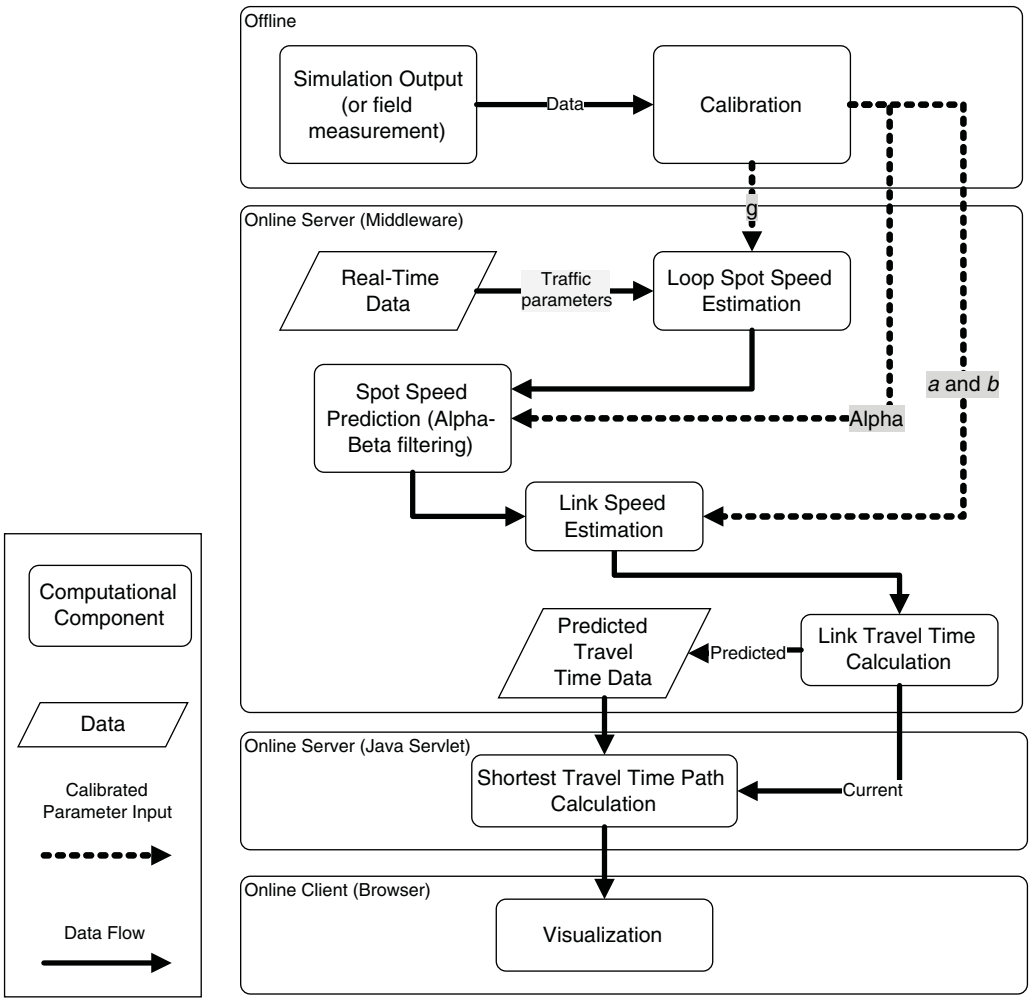


FIGURE 3 System design.

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. Loop spot speed S_L for time interval t is defined as

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

such that

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

where

N = interval traffic volume;
T = interval duration;
o = occupancy, or percentage of time a loop is occupied by vehicles per interval;
g = speed estimation parameter (*g*-factor), determined by effective vehicle length; and
L = mean effective vehicle length.

In some applications, g is considered time independent ($g = 2.4$) (*18*). For arterials, g is considered to be 2.63 (*17*). In the present application, $g = 2.14$, assuming the effective vehicle length is affected by transit and trucks (*19*). Equation 1 can be rewritten as

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

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

Prediction of Loop Spot Speed

Many models of speed prediction are developed and implemented online, such as the probability-based model by Lin et al. (*20*) and the knowledge-based model by Lee et al. (*21*). However, these models require many inputs, such as a signal timing plan. Moreover, a real-time system requires quick response and low computational cost.

The dynamic prediction of traffic parameters is suitable for this real-time application. Dynamic filtering techniques can not only smooth real-time data suffering from random errors but also predict data in the next state. Among dynamic filtering techniques, the Kalman filter has gained attention from system designers because it provides high prediction accuracy (*22*); many research projects have

demonstrated robustness and reliability for short-term traffic prediction in applications such as freeway speed (*23*, *24*), freeway travel time (*25*), and arterial travel time (*26*).

Guo et al. propose a Kalman filter-based method to predict freeway speed using data from single loop detectors (*23*). They assume that speed is the state of a discrete time-controlled processed governed by the linear stochastic difference equation as

$$S_L(t) = S_L(t-1) + e(t) \tag{3}$$

where $e(t)$ is the state process error with mean = 0 and variance Q. The empirical findings of Guo et al., $q(t)/o(t)$, indicate that the ratio of flow rate to occupancy (q/o) has a linear relationship with $S_L(t)$ (*23*). This linear relationship justifies the application of the Kalman filter. Thus, the measurement equation can be formulated as

$$\frac{q(t)}{o(t)} = gS_L(t) + \epsilon(t) \tag{4}$$

where

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

Next, the linear model (Equations 3 and 4) is solved by standard Kalman recursion equations, and the Kalman gain must be calculated recursively on the basis of calibrated g, Q, and R (*22*). Guo et al. adopted the smoothing function of the Kalman filter but neglected the prediction capability for the state variable because of the purpose of their research (*23*). When data are missing, $S_L(t)$ is supposed to be updated with a predicted value. However, the system state in Equation 3 cannot be updated because the equation lacks a term $u(t-1)$ to update the speed $S_L(t)$. [For more details about the Kalman filter, Welch and Bishop (*27*).] Moreover, variances R and Q usually must be calibrated with real data, and the calibration process will be tedious and cumbersome.

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

- The α-β filter has been widely applied to object tracking in image processing and can effectively predict the location of missing objects (*29–31*).
- Instead of using positions in the image, the α-β filter can mathematically predict speeds mainly on the basis of the q/o. The q/o measurement 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, α, 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 on the system.

In the study implementation, every single measurement $x(t) = q(t)/o(t)$ is smoothed and predicted. The α-β filter is defined in the following equations (*28*):

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

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

$$x_p(t+1) = \hat{x}\left(\frac{t+1}{t}\right) = x_s(t) + T \cdot v_s(t) \tag{7}$$

where

$x_s(t)$ = smoothed measurement at time stamp t,
$x_p(t)$ = predicted measurement at time stamp t,
α and β = fixed-coefficient filter parameters,
$x_o(t)$ = observed measurement (q/o) at time stamp t,
$v_s(t)$ = smoothed measurement changing rate (regarded as the velocity of the measurement) at time stamp t,
m = number of discrete time stamps since the last measurement, and
T = sampling interval (where $T = 1$ because data are updated every minute).

The filter starts with an initialization process defined by

$$x_s(1) = x_p(1) = x_o(1) \tag{8}$$

and

$$v_s(1) = 0$$

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

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

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

Equations 5 through 9 are used when the measurement can be input consistently into the system. As mentioned in the introduction, the data input could be missing because of communication errors or a measured speed of zero (when occupancy or flow rate = 0). In this case, the values of x and v can be predicted as

$$x_o(t) = x_s(t) = x_p(t) \tag{11}$$

and

$$v_s(t) = v_s(t-1)$$

Effectiveness of Prediction

Figure 4 shows the application of the α-β filter to data collected at the advance loop east of intersection 16 (Northeast 8th Avenue and 106th Avenue NE), westbound on Northeast 8th Avenue from 6 a.m.

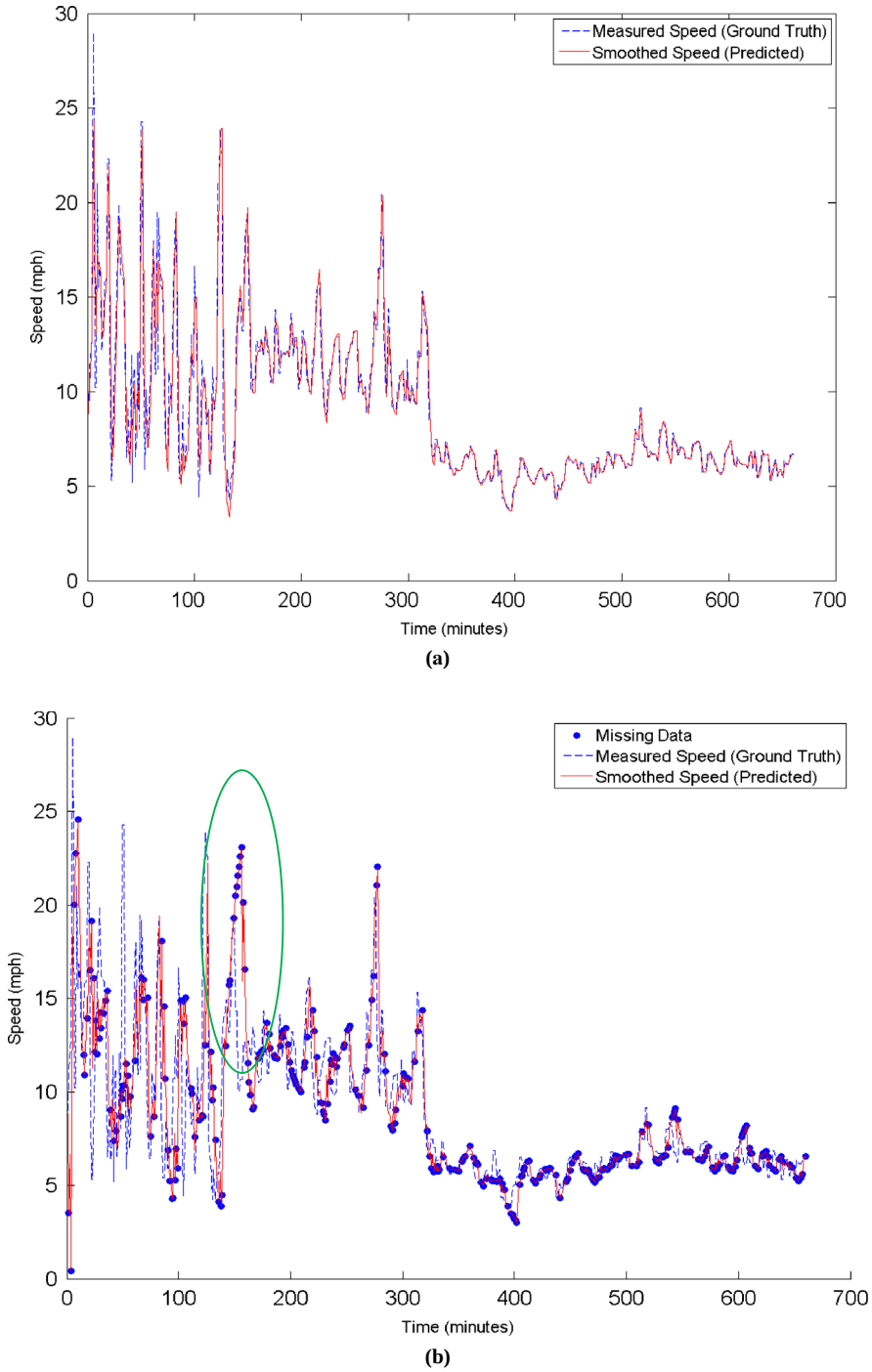


FIGURE 4 Application of α-β filter on loop spot speed prediction (α = 0.6): (*a*) no data missing and (*b*) 50% data missing.

to 7 p.m. on July 15, 2010. Figure 4*a* shows the effectiveness of filter smoothing, and Figure 4*b* shows the effectiveness of prediction when 50% of data are missing (randomly removed). The filter still smoothes the predicted measurement when data is missing.

However, the circled area of Figure 4*b* indicates that missing multiple data continuously would cause the filter to become increasingly inaccurate. This prediction constraint is common for most dynamic filters. In other words, dynamic filters can predict the trend in real time, but the object tends to be lost if abruptly turning or moving in another direction. However, the trend can be easily resumed once the true measurement enters the filter, as illustrated in the prediction results after the circled area.

Calibration of α

To improve prediction performance, the parameter α must be selected carefully. According to Equation 5, the higher α is, the more the fil-

ter will trust the correction from the new measurement. However, the system also 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 on the basis of the characteristics of each link, or one single α minimizing the system error can be adopted for the entire network. Either method 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

To quantify prediction performance, three measures of accuracy are used in this study (*33*):

$$\text{MAE} = \frac{\sum_{t=1}^{n} \left| F(t) - G(t) \right|}{n} \tag{12}$$

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

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

where

MAE = mean absolute error,
n = total number of samples,
$F(t)$ = predicted link travel at time interval t,
$G(t)$ = ground truth loop spot speed at time interval t, and
MAPE = mean absolute percentage error.

In the application, $F(t) = x_s(t)$. If data are missing, $x_o(t) = x_p(t|t-1)$ 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 show the relative improvement of the smoothing and effectiveness of prediction concurrently because $G(t)$ 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; it indicates the precision of prediction. MAE and RMSE can be evaluated jointly to determine the variation of the errors. Compared with RMSE, MAPE expresses the error as a percentage without exaggerating the error.

Parameter Calibration and Results

One day's worth of data collected on Thursday, July 15, 2010, were extracted from the database. Among 708 links, 472 links with advance detectors on the through-movement lanes were adopted in the RADAR Net system. Advance loop data on 23 major arterials links (average flow rate > 400 vehicles per hour and average occupancy > 20%) were selected for calibration and evaluation. Data before 6 a.m. and after 7 p.m. were excluded from the data set because few traffic fluctuations are observed during these periods (in other words, including the data could result in underestimating average prediction errors).

To determine the most suitable α and effectiveness of prediction, different percentages of data (10% to 90%, in 10% increments) were removed at random from each data set. Results are illustrated in Figure 5, sequentially from bottom to top. If no data are missing, α = 0.9 can result in the "best" results, which is not surprising because the filter "trusts" the new measurement more (Figure 5*a*).

The ground truth measurements may contain random errors; hence, the ground truth is not the "real" ground truth. In this case, low MAE, RMSE, and MAPE may not absolutely imply that the filter performs better. Therefore, the α = 0.9 case in Figures 5*a*, 5*b*, and 5*c* may imply that the filter is affected by the noise. In contrast, the α = 0.1 case shows the "worst" results, which also is reasonable because the filter does not "trust" the new measurement.

Figure 5*c* shows a decreasing trend in MAPE, indicating that the percentage error is reduced when the filter trusts the measurements more. It shows two drops at α = 0.2 and α = 0.4, indicating that these two values could be used if the data quality were poor. Overall, the α = 0.6 case shows drops in both MAE and RMSE when 90% of the data are missing, which implies that α = 0.6 can reduce noise and smoothly predict results concurrently. When 80% of the data are missing, the drop also appears in MAE and RMSE. However, as α increases, the filter becomes more sensitive to noise. Hence, α = 0.8 may be used for the links with higher-quality data and low percentage of missing data. In the study application, α = 0.6 was selected because this value showed its robustness to consecutively missing data and could minimize random errors when measurements were available.

Practical Constraint and Remedy

The loop-based methods can perform accurately under congested conditions (e.g., *17, 23*); however, overnight, the loop-based method would result in incorrect results as a result of low flow rate and occupancy. Therefore, a threshold value was required to differentiate congested and uncongested conditions. Guo et al. (*23*) and Coifman (*34*) recommended 10% occupancy as an optimal threshold value, and this value is used in the study system.

Link Speed Estimation

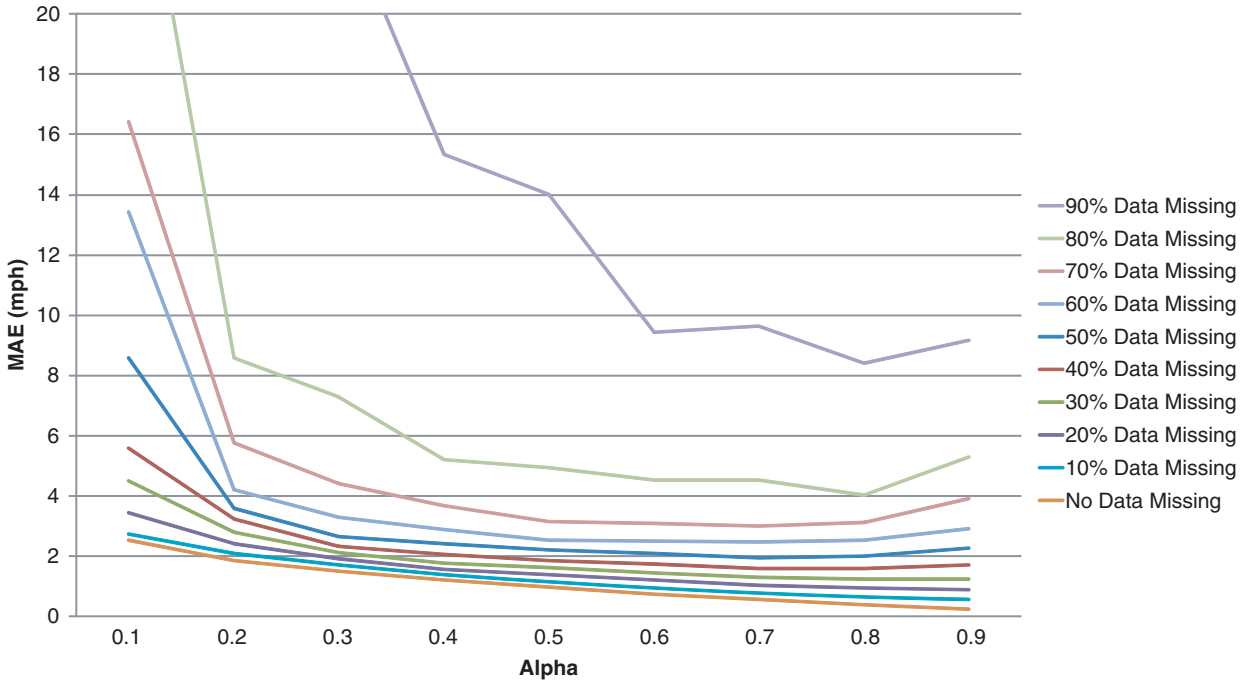
Once estimated and predicted, loop spot speeds can be converted to link speeds to better represent link performance. According to Zhang, loop spot speed can represent link speed under congested conditions (*17*). Wu et al. found that advance loop spot speed is likely to overestimate link speed if the ground truth link speed is higher (*19*). Hence, the model to represent the relationship between ground truth link speed and advance loop spot speed is formulated as (*19*)

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

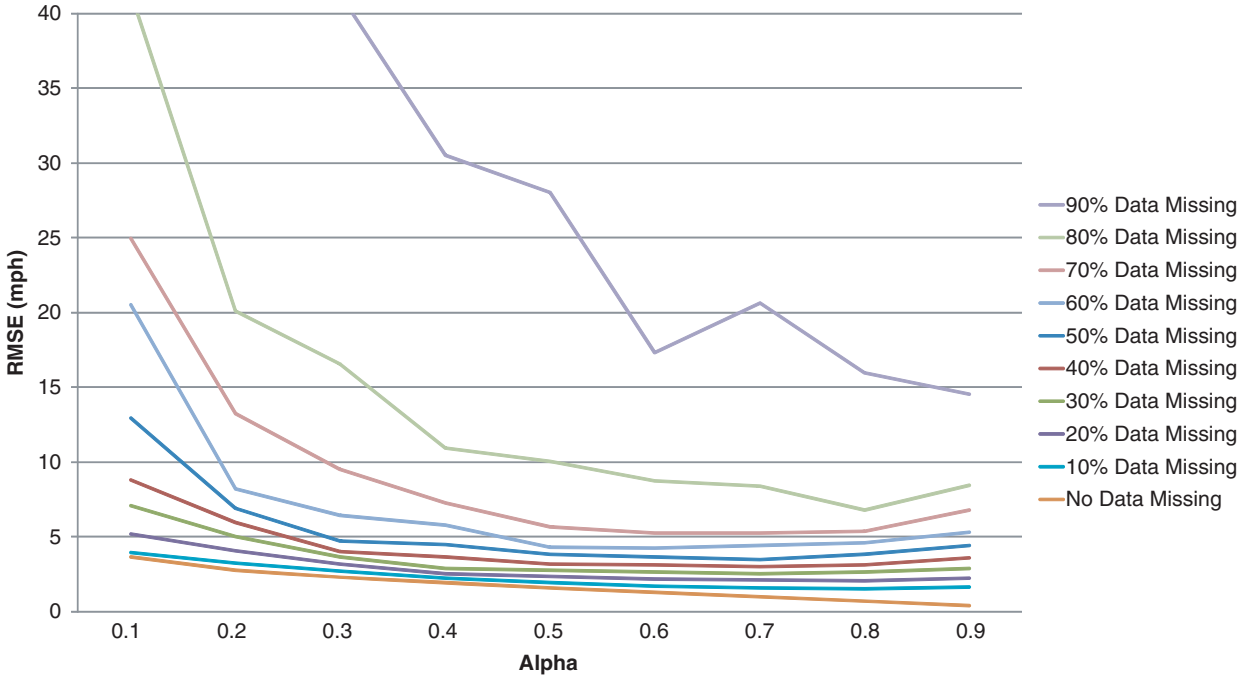
where

$\hat{S}_J(t)$ = estimated link speed,
a and b = coefficients that require calibration, and
$\hat{S}_L(t)$ = loop spot speed at the advance detector.

The constant value of 1 allows the calibrated model to traverse the origin (0,0). In other words, when $\hat{S}_L(t) = 0$, $\hat{S}_J(t) = 0$, assuming that



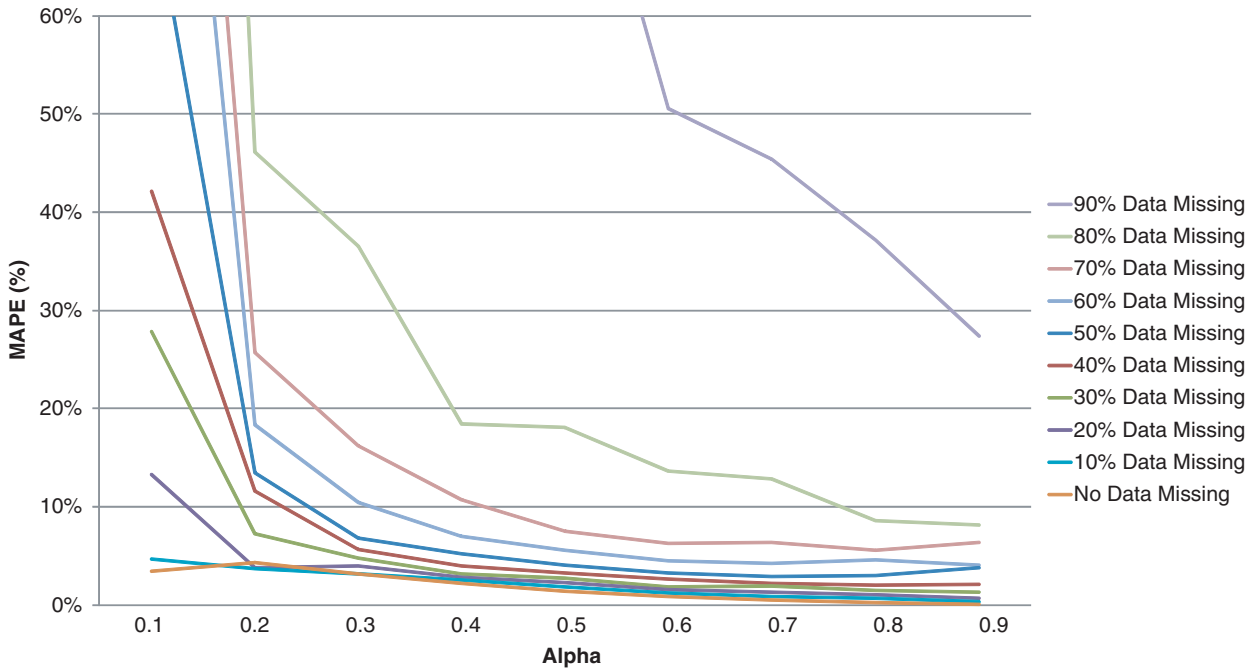
(a)



(b)

FIGURE 5 Optimal α-value selection based on percentages of missing data: (*a*) relationship between α and MAE, (*b*) relationship between α and RMSE.

(continued)



(c)

FIGURE 5 *(continued)* **Optimal α-value selection based on percentages of missing data: (*c*) relationship between α and MAPE with different percentages of missing data.**

the spot speed should be equal to the link speed when the measured spot speed is close to zero (*17*).

As implemented, the universal parameters *a* and *b* are calibrated by the VISSIM traffic simulation software package as 1.0 and 0.05, respectively, on the basis of the major streets and applied to all links to demonstrate the feasibility of the approach (*35*). To achieve the most accurate results, calibration should be conducted for every link. After link speeds are available, link travel time can be calculated easily with known link lengths. Because estimating link travel speed takes into account signal control, the calculated link travel time also includes the control delay (*19*). Therefore, route travel time is simply the summation of all links along a route.

SYSTEM IMPLEMENTATION

Implementation

According to the system design illustrated in Figure 3, the online client and online server (Java Servlet) layers of the RADAR Net system are programmed using the Google web toolkit (*36*) combined with Eclipse, an open-source Java integrated development environment (*37*). Development efficiency has been greatly improved over the previous development environment for the GATI system. The code is optimized and converted to JavaScript code by the Google web toolkit. The online server layer is implemented in C#. The server is equipped with two Intel Xeon E5520 (2.26-gigahertz, 4 cores/8 threads) central processing units with 24 gigabytes of DDR3 random-access memory. The server runs on a Windows Server 2008 operating system with MS Structured Query Language Server 2008.

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-path routing, and
5. Arterial data sharing.

Modules 1 through 3 were the reimplementation based on the GATI system with improved database design and performance (*9*).

Figure 6*a* is the flow rate–occupancy scatter plot for June 20–26, 2010; statistical analysis is performed online by clicking the appropriate button. Figure 6*b* is the time–domain plot of flow rate–occupancy data for the same period. The scalable visualization bar easily zooms in to June 23, and the window slides to investigate traffic variation. Figure 6*c* is the real-time traffic map; one can see that conditions are not free flowing on all links between Intersections A and B. Figure 6*d* shows the shortest path between Intersections A and B. The shortest path is calculated by the A* algorithm with real-time link speeds updated by the α-β filter (*38*). Because speed data are stored in the database, the algorithm can be executed in real time when two nodes are selected; the path successfully skips the congested links shown 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 module for arterial data analysis to investigate causes of the bottleneck.

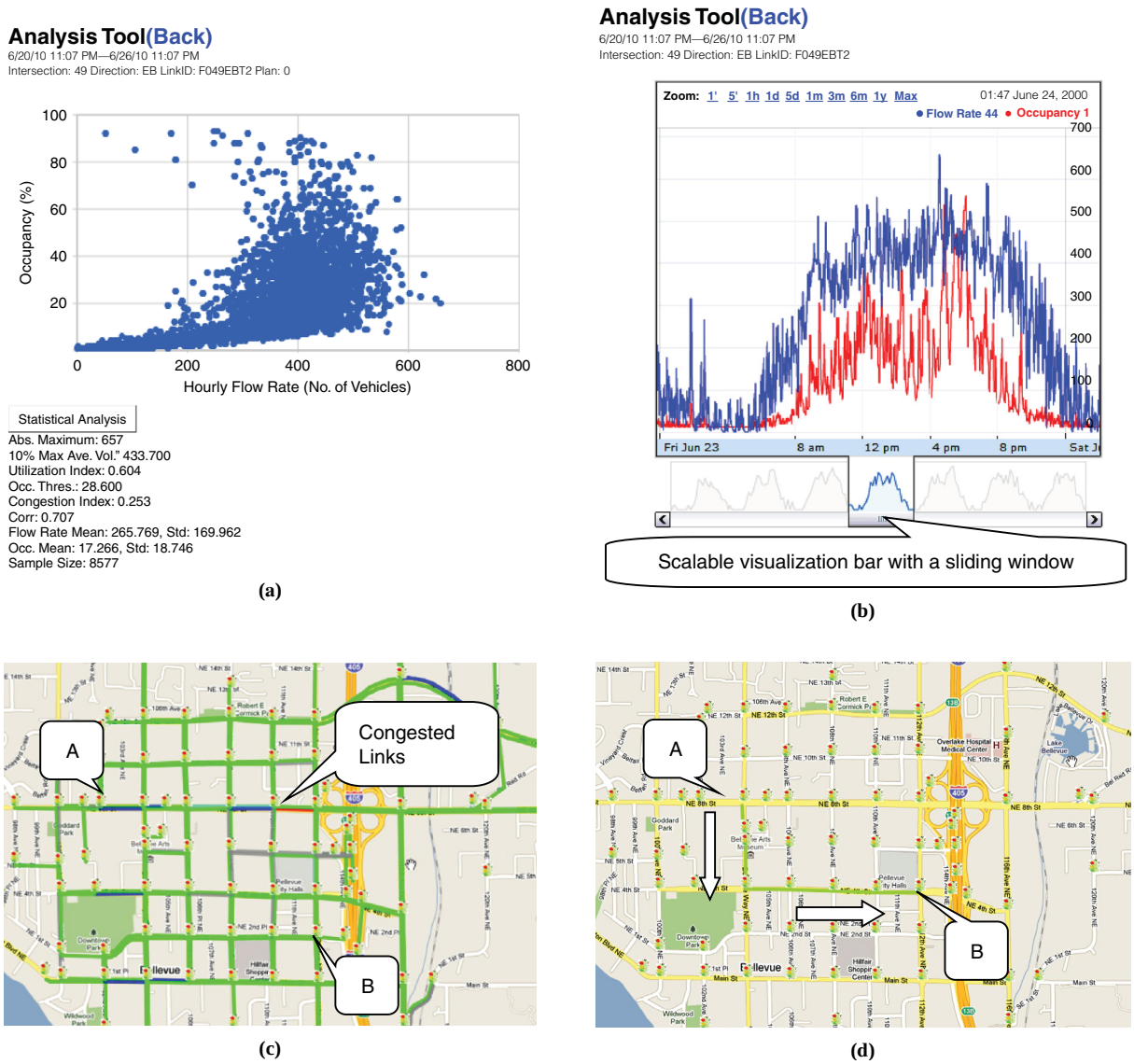


FIGURE 6 RADAR Net modules: (*a*) volume (as flow rate)–occupancy scatter plot and analysis (June 20–26, 2010), (*b*) scalable time–domain plot (June 20–26, 2010), (*c*) real-time traffic flow map at 5:45 p.m. on Thursday, July 29, 2010, and (*d*) real-time dynamic shortest (travel time) path routing.

Performance

The database design effectively reduces the query time for estimating loop spot speed and meanwhile increases the online algorithm performance. A query for all attributes of the entire network by joining all tables is retrieved in less than 500 ms. Downloading raw data, calculating link speeds, and updating travel time data for all links in the network takes less than 2 s, on average, in the middleware. Moreover, the shortest-path algorithm can be calculated in 1 s, on average, within the central business district.

CONCLUSIONS AND RECOMMENDATIONS

System performance and simple implementation are the keys to the success of a real-time decision-making system for a large urban arterial network. Many practical challenges present hurdles to the development of such a system. This paper demonstrates the computational capability provided by the multitier architecture of the web-based RADAR Net system. A practical and scalable arterial database design also is proposed. The schemas can be used as templates for storing arterial data for other agencies. On the basis of the relational model, the database design can incorporate more arterial data from other sources and increase system scalability.

RADAR Net's four-layer design successfully distributes the computational burden. Traffic parameters are calculated or retrieved directly from the loop detector. The system can dynamically predict and smooth real-time loop spot speeds with the use of an α-β filter (a simplified version of the Kalman filter) while maintaining high system performance. Many application modules based on the system architecture have been implemented and have proven the system feasible for performing real-time analysis and assisting with decision making.

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

Even though it demonstrates an ability to process real-time data, the RADAR Net system has some limitations. For example, the prediction function cannot address long-term missing data or malfunction in loop detectors. For cities without loop detector infrastructure or with detectors missing in some roadway links, advanced sensor technologies such as Bluetooth travel time detectors can provide missing travel time data. Moreover, performance may be reduced if thousands of queries are executed simultaneously. One possible solution is to use concurrency control. In addition, database design could be further improved by incorporating multidimensional databases into the system to handle aggregated data in real time. Cloud computing could be a potential means of increasing the computing power of RADAR Net.

ACKNOWLEDGMENTS

The authors thank Fred Liang, Mike Whiteaker, and Brook Durant of the City of Bellevue, Washington, for providing arterial traffic data. Special thanks go to Mengqi Wu for coding the first generation of RADAR Net and to 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.* Center for Transportation Analysis, Oak Ridge National Laboratory, Oak Ridge, Tenn., 2002.
2. Seymour, E. J., and B. Miller. Use of A Google-Based Mapping System for Traffic Management Center Web Navigation. Presented at 86th Annual Meeting of the Transportation Research Board, Washington, D.C., 2007.
3. Chen, C. *Freeway Performance Measurement System (PeMS).* Publication UCB-ITS-PRR-2003-22. California Partners for Advanced Transit and Highways (PATH), University of California, Berkeley, Richmond, Calif., 2003.
4. Bertini, R. L., S. Hansen, A. Byrd, and T. Yin. Experience Implementing a User Service for Archived Intelligent Transportation Systems Data. In *Transportation Research Record: Journal of the Transportation Research Board, No. 1917,* Transportation Research Board of the National Academies, Washington, D.C., 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. Presented at 87th Annual Meeting of the Transportation Research Board, Washington, D.C., 2008.
6. Petty, K., J. Kwon, and A. Skabardonis. APeMS: An Arterial Performance Measurement System. Presented at 85th Annual Meeting of the Transportation Research Board, Washington, D.C., 2006. http://www.signalsystems.org.vt.edu/documents/Jan2006AnnualMeeting/SundayWorkshop/Petty_apems_trb.pdf.
7. Cheevarunothai, P., Y. Wang, and N. L. Nihan. Identification and Correction of Dual-Loop Sensitivity Problems. In *Transportation Research Record: Journal of the Transportation Research Board, No. 1945,* Transportation Research Board of the National Academies, Washington, D.C., 2006, pp. 73–81.
8. Wu, Y.-J., Y. Wang, and D. Qian. A Google-Map-Based Arterial Traffic Information System. *Proc., ITSC 2007: IEEE Intelligent Transportation Systems Conference,* Oct. 22, 2007, Seattle, Wash., IEEE, Piscataway, N.J., 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. *Communications of the ACM,* Vol. 13, No. 6, 1970, pp. 377–387.
11. Garcia-Molina, H., J. D. Ullman, and J. Widom. *Database Systems: The Complete Book,* 2nd ed. Pearson Prentice Hall, Upper Saddle River, N.J., 2009.
12. Ma, X., Y.-J. Wu, and Y. Wang. DRIVE Net: E-Science Transportation Platform for Data Sharing, Visualization, Modeling, and Analysis. In *Transportation Research Record: Journal of the Transportation Research Board, No. 2215,* Transportation Research Board of the National Academies, Washington, D.C., 2011, pp. 37–49.
13. Klein, L. A. *Sensor Technologies and Data Requirements for ITS.* Artech House, Boston, Mass., 2001.
14. Klein, L. A., M. K. Mills, and D. R. P. Gibson. *Traffic Detector Handbook.* Turner–Fairbank Highway Research Center, FHWA, U.S. Department of Transportation, McLean, Va., 2006.
15. Athol, P. Interdependence of Certain Operational Characteristics Within a Moving Traffic Stream. In *Highway Research Record 72,* HRB, National Research Council, Washington, D.C., 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. In *Transportation Research Record: Journal of the Transportation Research Board, No. 1676,* TRB, National Research Council, Washington, D.C., 1999, pp. 109–115.
18. Ishimaru, J. M., and M. E. Hallenbeck. *Flow Evaluation Design Technical Report.* Publication WA-RD 466.2. Washington Department of Transportation, Olympia, 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. Presented at 90th Annual Meeting of the Transportation Research Board, Washington, D.C., 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, 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. In *Transportation Research Record: Journal of the Transportation Research Board, No. 1879,* Transportation Research Board of the National Academies, Washington, D.C., 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. In *Transportation Research Record: Journal of the Transportation Research Board, No. 1959*, Transportation Research Board of the National Academies, Washington, D.C., 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 1, 2010.
28. Blackman, S. S. *Multiple Target Tracking with Radar Applications.* Artec House, Norwood, Mass., 1986.
29. Malinovskiy, Y., Y.-J. Wu, and Y. Wang. Video-Based Monitoring of Pedestrian Movements at Signalized Intersections. In *Transportation Research Record: Journal of the Transportation Research Board, No. 2073*, Transportation Research Board of the National Academies, Washington, D.C., 2008, pp. 11–17.
30. Wu, Y.-J., F.-L. Lian, and T.-H. Chang. Traffic Monitoring and Vehicle Tracking Using Roadside Cameras. *Proc., 2006 IEEE International Conference on Systems, Man and Cybernetics, Volume 6,* Taipei, Taiwan, 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, pp. 513–520.
32. Kalata, P. R. The Tracking Index: A Generalized Parameter for α-β and α-β-γ Target Trackers. *IEEE Transactions on Aerospace and Electronic Systems*, Vol. AES-20, No. 2, 1984, pp. 174–182.
33. Washington, S. P., M. G. Karlaftis, and F. L. Mannering. *Statistical and Econometric Methods for Transportation Data Analysis.* Chapman & Hall/CRC, Boca Raton, Fla., 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. *VISSIM User Manual: Version 5.2.* PTV, Karlsruhe, Germany, 2010.
36. Google, Inc. *Google Web Toolkit.* http://code.google.com/webtoolkit/. Accessed July 29, 2010.
37. The Eclipse Foundation. Eclipse. Ottawa, Ontario, Canada. 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. *IEEE Transactions on Systems Science and Cybernetics*, Vol. 4, No. 2, 1968, pp. 100–107.

The Information Systems and Technology Committee peer-reviewed this paper.